

HYBRID CONTROL STRATEGIES FOR RAPID, LARGE ANGLE SATELLITE SLEW MANEUVERS

THESIS

David B. French, Captain, USAF

AFIT/GA/ENY/03-2

DEPARTMENT OF THE AIR FORCE
AIR UNIVERSITY
AIR FORCE INSTITUTE OF TECHNOLOGY

Wright-Patterson Air Force Base, Ohio

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

The views expressed in this thesis are those of the author and do not reflect the official policy or position of the United States Air Force, Department of Defense, or the U. S. Government.

AFIT/GA/ENY/03-2

HYBRID CONTROL STRATEGIES FOR RAPID, LARGE ANGLE SATELLITE SLEW MANEUVERS

THESIS

Presented to the Faculty

Department of Aeronautical and Astronautical Engineering

Graduate School of Engineering and Management

Air Force Institute of Technology

Air University

Air Education and Training Command

In Partial Fulfillment of the Requirements for the

Degree of Master of Science in Astronautical Engineering

David B. French

Captain, USAF

March 2003

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

AFIT/GA/ENY/03-2

HYBRID CONTROL STRATEGIES FOR RAPID, LARGE ANGLE SATELLITE SLEW MANEUVERS

David B. French
Captain, USAF

Approved:

Richard G. Cobb, Maj, USAF (Advisor)	12 MAR 03 date
Dr. Steven G. Tragesser	12 MAR 03 date
Dr. Bradley S. Liebst	12 MAR 03 date

Acknowledgements

I would like to thank Maj Rich Cobb for his expert instruction, patience, and for keeping his sense of humor. Thanks to Wilber Lacy for invaluable technical assistance and to Jay Anderson for funding my efforts. Thanks to Giampiero Campa of West Virginia University for trying to help a stranger—with no experience programming in C—debug his C-code. Thanks to Tim Hart of I-70 Paintball for a practical education on paintball gear. Thanks to Vincent Dabrowski for all the work you put into nursing SIMSAT back from the brink and then helping me figure it out. Thanks to Lee Philley for teaching me the difference between a dove and a thrush and for bailing me out late on Sunday night many times. Thanks to Joel Witte for being a good friend, a good teacher and a good sport while the butt of all our Top Gun jokes. Thanks to Ryan Quaale for pointing out that Joel had descended below the hard deck before making the kill, to Brian Foster for allowing me to get all his sleep for him and to Troy Tschirhart for being a good boss and "keeping it real." Thanks to Dr. Steve Tragesser for teaching me the difference between body rates and Euler rates—they are different, aren't they? Thanks to Bishop the beagle dog for only pooping on the carpet twice while I was at school and to the Machine Shop for magically turning poop on paper into metal parts. Thanks to C. Eugene Skelton II, if that is your real name, for helping me get started. Thanks to Dr. Wiesel for answering many questions unworthy of his intelligence and education, and to Mom and Dad for teaching me what is truly important. Finally, my love and thanks go to my fiancée for her support and for agreeing to put up with me for the rest of my life.

Table of Contents

Page

Acknowledgements iv

Table of Contents v

List of Figures viii

List of Tables xi

List of Abbreviations xii

List of Abbreviations xii

List of Symbols xiii

Abstract xv

I. Introduction 1

The Past 1
The Present 3
The Future 5
Objectives 6
Methodology 8

II. Literature Review 9

Present Day Control Techniques 9
Non-linear control algorithms 9
Lyapunov Stability Techniques 9
Two-point boundary value problems 10
State Dependent Riccati Equation 11
Precision Pointing 12
Similar Efforts 13
SADSaC 13

Page

III. Design 14

Overview 14
SIMSAT 14
Physical Properties 15
Interface 16
Control 16
Computer Simulation of SIMSAT 17
Reference Frames 17
Plant Modeling 21
Control 25
PD Control 25
SDRE Control 27
Control Switching 32
Thruster Installation 34
Reaction Wheel Notes 41

IV. Results 42

Overview 42
Experiment Process 42
Preparation 42
Testing 43
Closed-Loop Reaction Wheel Control 44
Open-Loop Thrust 46
Closed-Loop Thrust Control 48
Hybrid Control 49
Control Switching 49
Fine Pointing 53
Model Matching 54
Constant Gains – Varied Angle Matching 58
Constant Angle – Varied Gains Matching 64
Maximum Performance Study 68
SDRE/LQR Control 71
SIMSAT Computer Model Animation 74

V. Conclusions and Recommendations 76

Conclusions 76
Recommendations 77
Action Items 77
Future Study 79

Appendix A: Simulink Models 80

Page

Appendix B: MATLAB® Code 96

Appendix C: Data Tables 114

Bibliography 115

Vita 118

List of Figures

Page

Figure 1. Pneumatic attitude control. 2

Figure 2. Hubble Space Telescope 5

Figure 3. SIMSAT 7

Figure 4. SIMSAT prior to modifications 15

Figure 5. SIMSAT reaction wheels. 17

Figure 6. SIMSAT shown with body-fixed axes. 18

Figure 7. Body frame points of reference (top view). 18

Figure 8. 3-2-1 rotation sequence. 19

Figure 9. Plant Model 24

Figure 10. Scalable reaction wheel dynamics. 24

Figure 11. Example PD Controller 25

Figure 12. Energy curve and components for single-axis maneuver 34

Figure 13. PMI® Pure Energy 88 cubic inch, 4500 psi Fixed Nitrogen System 35

Figure 14. PMI Pure Energy THOR Regulator. 36

Figure 15. Nitrogen solenoid valves. 37

Figure 16. SIMSAT input/output panel. 37

Figure 17. Relays mounted on circuit board. 38

Figure 18. Original design thruster positions. 39

Figure 19. Thruster system layout on first radial plate. 41

Figure 20. SIMSAT pitch/yaw restraints. 43

Figure 21. ControlDesk layout 44

Figure 22. Initial Results for reaction wheel matching. 45

Page

Figure 23. Matched wheel response. 46
Figure 24. Open-loop thrust command response. 47
Figure 25. Closed-loop thrust control. 49
Figure 26. Control switching 50
Figure 27. Energy Switch Study. 51
Figure 28. Normal handoff from thrusters to wheels 52
Figure 29. Worst case pointing error. 53
Figure 30. Thrust matching assuming symmetric thrust. 55
Figure 31. Potential cause for thruster bias. 56
Figure 32. Thrust matching including 25 percent negative bias 57
Figure 33. Constant Gain vs. Varied Angles 58
Figure 34. Displacement of 20 degrees. 59
Figure 35. Negative 20-degree rotation 60
Figure 36. Positive 40-degree rotation. 61
Figure 37. Negative 40-degree rotation 62
Figure 38. Positive 60-degree rotation. 62
Figure 39. Negative 60-degree rotation 63
Figure 40. Positive 80-degree rotation. 64
Figure 41. Constant Angle – Varied Gain 65
Figure 42. Thruster Rate Gain of 10. 66
Figure 43. Thruster Rate Gain of 15. 66
Figure 44. Thruster Rate Gain of 20 67

Page

Figure 45. Thruster Rate Gain of 25. 67

Figure 46. Thruster Rate Gain of 30. 68

Figure 47. Settling Time vs. Fuel Expenditure..... 69

Figure 48. Minimum settling time vs. rotation angle..... 70

Figure 49. Results of SDRE control applied to SIMSAT computer model. 72

Figure 50. Limited relationship between PD rate gain and SDRE Q matrix 74

Figure 51. SIMSAT computer model animation screen shot. 75

List of Tables

Page

Table 1. Attitude and orbital control systems. 3

Table 2. Control method vs. typical pointing accuracy (de Weck, 2001). 12

Table 3. Sample data for thrust vs. fuel expenditure tests. 69

Table 4. Constant gain – varied angle data 114

Table 5. Constant angle – varied gain data excerpt 114

List of Abbreviations

AFIT	Air Force Institute of Technology
CSEL	control switch energy level
DOA	domain of attraction
FEP	fluorinated ethylene propylene
HST	Hubble Space Telescope
LQR	linear quadratic regulator
PD	proportional differential
PI	proportional integral
SADSaC	Satellite Attitude Dynamics Simulation and Control
SDRE	State Dependant Riccati Equation
SIMSAT	simulated satellite
TPBVP	two-point boundary value problem

List of Symbols

$\delta\Omega$	Change in reaction wheel speed
$\Delta\theta_1$	Difference between desired and actual 1st Euler rotation angle
$\Delta\theta_2$	Difference between desired and actual 2nd Euler rotation angle
$\Delta\theta_3$	Difference between desired and actual 3rd Euler rotation angle
θ_1	1st Euler rotation angle
θ_2	2nd Euler rotation angle
θ_3	3rd Euler rotation angle
ω_0	Initial angular rate
ω_1	Body-frame rotation rate about axis 1
ω_2	Body-frame rotation rate about axis 2
ω_3	Body-frame rotation rate about axis 3
ω_f	Final angular rate
Ω_1	Angular speed of reaction wheel rotating about body axis 1
Ω_2	Angular speed of reaction wheel rotating about body axis 2
Ω_3	Angular speed of reaction wheel rotating about body axis 3
A	System dynamics matrix for LQR and SDRE
B	Control dynamics matrix for LQR and SDRE
b_1	First axis of body frame
b_2	Second axis of body frame
b_3	Third axis of body frame
C	Observation matrix for LQR and SDRE
d_1	Distance between center of gravity and thruster acting about body axis 1
d_2	Distance between center of gravity and thruster acting about body axis 2
d_3	Distance between center of gravity and thruster acting about body axis 3
D	Feed forward matrix for LQR and SDRE
E	System “energy”
i_1	First axis of inertial frame
i_2	Second axis of inertial frame
i_3	Third axis of inertial frame
I_{11}	Mass moment of inertia about axis 1
I_{22}	Mass moment of inertia about axis 2
I_{33}	Mass moment of inertia about axis 3
I_w	Mass moment of inertia for reaction wheel
J	Cost function
k_T	Shaping factor for “kinetic energy”
k_V	Shaping factor for “potential energy”
K	Generic gain
K_d	Derivative gain for PD controller
K_{lqr}	LQR-generated gain
K_{motor}	Scaling factor multiplied by reaction wheel hysteresis curve output
$K_{motor,in}$	Scaling factor multiplied by reaction wheel hysteresis curve input
K_p	Proportional gain for PD controller
k_{scale}	Factor multiplied by thruster rate gain to calculate wheel rate gain
M_1	Moment about body axis 1

M_2	Moment about body axis 2
M_3	Moment about body axis 3
Q	State weighting matrix
R	Control weighting matrix
Q_T	State weighting matrix for thrusters
R_T	Control weighting matrix for thrusters
Q_W	State weighting matrix for reaction wheels
R_W	Control weighting matrix for reaction wheels
R_{ib}	Rotation matrix from inertial to body frame
t_0	Initial time
t_f	Final time
$t_{f,act}$	Final time for actual SIMSAT
$t_{f,mod}$	Final time for computer modeled SIMSAT
T	System "kinetic energy"
T(-)	Negative yaw thrust
T(+)	Positive yaw thrust
T_1	Thrust acting to create moment about body axis 1
T_2	Thrust acting to create moment about body axis 2
T_3	Thrust acting to create moment about body axis 3
u	Control vector
V	System "potential energy"
x	State vector
y	Observed vector

Abstract

This research investigated hybrid control strategies for rapid satellite pointing. First, a detailed computer simulation model of AFIT's SIMSAT satellite simulator was constructed. Control strategies were developed to enable the system to perform large-angle, 3-axis slewing maneuvers using a combination of both thrusters and reaction wheels. To handle the non-linear model, a State Dependent Riccati Equation controller was programmed and successfully controlled the computer-modeled satellite for any given slewing maneuver. A simpler PD controller was then programmed and demonstrated on the computer simulation of SIMSAT, using a combination of thruster and reaction wheel control inputs for large-angle single axis maneuvers and for small angles using three-axis control. There was good agreement between the experimentally obtained maneuver results and those produced with the computer simulation model for the single-axis case. Lastly, the trade-off between settling time and thruster fuel is discussed, as well as the variation of gains required to achieve maximum performance for a desired slew.

HYBRID CONTROL STRATEGIES FOR RAPID, LARGE ANGLE SATELLITE SLEW MANEUVERS

I. Introduction

The Past

Since October 4, 1957, satellites have patrolled the vacuum beyond earth's atmosphere. Sputnik I introduced the world to a new era of technological capabilities by proving man could successfully exploit space. Its mission was simple—merely broadcasting its position via a series of beeps to 80 percent of the world (Soviet Satellites, 1968: 174). However, it wasn't long until both the Soviet Union and its rival, the United States, began to expand on this leap. Ideas developed in the years leading up to Sputnik began to be implemented. The US launched the Vanguard satellites, designed to allow ground stations to track it and infer information on the Earth's shape and gravitational field. Then came the Explorer series, which first detected, then confirmed the existence of the Van Allen Belts. Pioneer, Luna and scores of other satellite programs followed, each paving the way for a new facet of satellite exploration of Earth, extraterrestrial bodies and space, as well as the potential for manned space flight.

With this growth in function came a requirement for increased complexity in many areas—among them, spacecraft attitude control. Expanding satellite mission requirements introduced the need to orient spacecraft precisely, whether for the purpose of pointing onboard instruments or orienting spacecraft in preparation for reentry through narrow safe corridors into the earth's atmosphere. The first attitude control systems were built for this purpose. They used solid propellant and provided needed external torques for Vostok and Mercury spacecraft.

However, even before Sputnik I was launched, men had conceptualized a wide variety of other means of attitude control. Many were quite close to systems in use today—others weren't.

One design, proposed in 1954, utilized a pneumatic system (Figure 1) and was described as such:

> a, b, c and d are bleeding nozzles, whereas pressure air is supplied. h is a segment, mounted on the axle of the top casing of the rate gyro in question, and h' a vane mounted on the axis…of the angular acceleration detector. h' is acted upon at both ends by the air jets of c and d. If these jet forces differ, a torque is acting on h'.

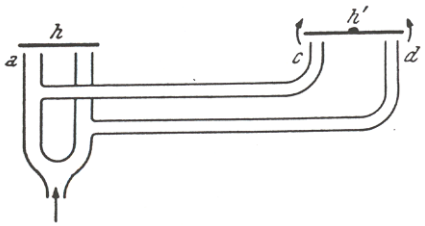


Figure 1. Pneumatic attitude control.

This resulting torque on h' would accelerate the vane and the resulting angular momentum would counteract the original disturbance. (Kooy, 1954: 2). While this was a crude system, it reveals the span of ideas early designers were considering for attitude control. It also infers that designers had considered the limitations of the earliest of attitude control schemes—thrusters.

In the early years after Sputnik, however, most attitude control was accomplished by thruster systems. This changed when a new technology, the photovoltaic cell, was developed by Bell Labs in 1954 (Thomas, 1994: 2). The photovoltaic cell allowed for the development of solar panels, which would soon after provide satellite designers with an alternative to thruster systems. The power harnessed from the sun would drive

electrically powered attitude control devices such as momentum wheels, control moment gyros and electro-magnets. This provided several benefits, including potential for extended operating life.

The Present

Satellites today are launched for a wide variety of purposes. Few purposes allow for the spacecraft to be completely free of attitude control. Table 1 (Wertz, 2001: 6) shows both orbital control and attitude control requirements of a variety of mission types. Since this paper focuses mainly on attitude control, that column is of particular interest.

Mission Type	Field*	Typical Orbit Requirement	Typical Attitude Requirement
Communications, Broadcasting	Com/Mil	Geosynchronous, low-Earth constellation	Antenna pointing for Earth coverage
Space manufacturing	Com	Any available at low cost	Very small disturbances
Space burial	Com	Any low cost	Any
Earth observations	Com/Sci/Mil	Orbit control not critical, but precise position knowledge required	High precision
Data store and forward	Com/Sci	Low-Earth orbit with minimal requirements	Little or no requirements
Space interferometer	Sci	Very high precision formation flying	Mid-precision
Space telescope	Sci	Minimal requirements	Highest available precision
Interplanetary exploration	Sci	Precision targeting	Camera and antenna pointing
Global Positioning System	Mil/Civ	Very high precision knowledge	Moderate
Space-based laser	Mil	High precision knowledge needed	Very high precision

* Com = Commercial, Sci = Scientific, Mil = Military, Civ = Civilian

Table 1. Attitude and orbital control systems.

As the table illustrates, the requirement for attitude control varies widely depending on the mission requirements. How do spacecraft meet these requirements?

Specific examples of modern satellites will serve to illustrate how a wide variety of controls are employed.

Loral's Intelsat 7 is a communications satellite. Its mission is to receive and transmit focused beams of electromagnetic waves. It uses three-axis control via thrusters for control to *assist* in control of its "independently steerable, high-powered, Ku-band spot beams" (Intelsat 7,7A). The French communications satellite Telecom and Loral's Globalstar communications satellites also use three-axis thruster control for similar reasons—it requires antennae pointing, but not to a high degree of precision.

On the other hand, the Hubble Space Telescope (Figure 2) uses four relatively small reaction wheels to turn the telescope. For the purposes of the spacecraft, the slew rate of 90 deg in 20 minutes is not a mission impediment. The benefit of this low wheel inertia system is that it allows the telescope to be pointed extremely precisely—a critical requirement. "In order to take images of distant, faint objects, Hubble must be extremely steady and accurate. The telescope is able to lock onto a target without deviating more than 7/1000th of an arcsecond, or about the width of a human hair seen at a distance of 1 mile" (Quick Facts, 2003). Reaction wheels have a tendency to saturate over time and therefore need a method of desaturation—typically thrusters. However, Hubble does not use thrusters due to possible optics contamination. It uses magnetorquers to unload momentum from the reaction wheels (Jane's, 2002).

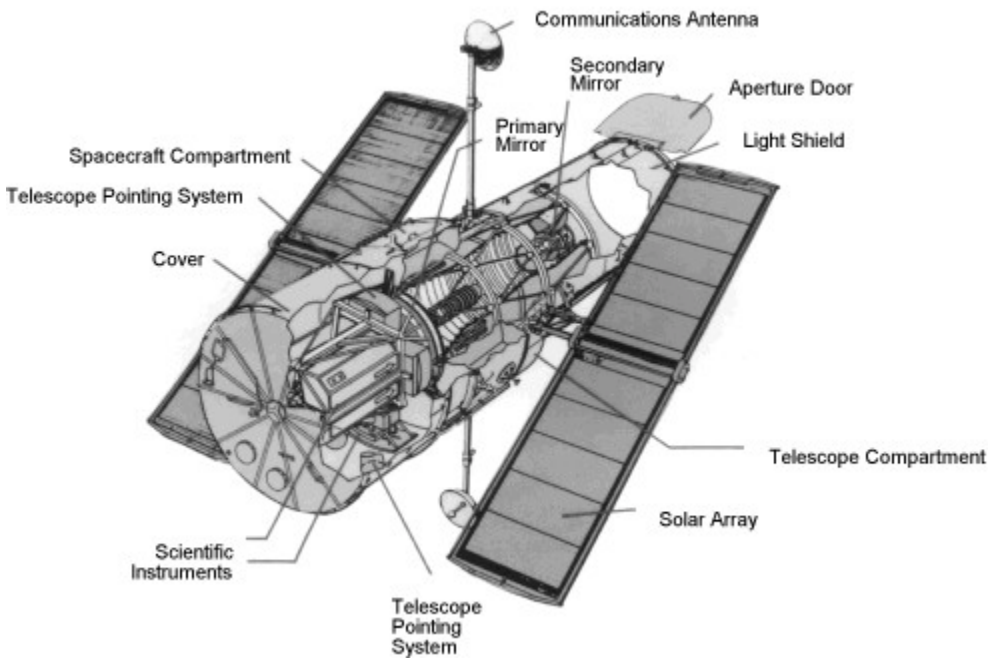


Figure 2. Hubble Space Telescope

The Cassini space probe does use the combination of thrusters and reaction wheels. With this system, it maintains 2 milliradian pointing accuracy for its array of planetary observation equipment.

Finally, Intelsat 901, a communications satellite, uses 2 momentum wheels and magnetic torquers to achieve a pointing accuracy of between 0.006 and 0.305 degrees pointing accuracy (Wertz, 2001: 13).

The Future

Referring back to Table 1, the last entry refers to a "Space based laser," a future application that will require not only high precision pointing as noted, but higher performance maneuvering as well (i.e. faster slew rates). It is this combination that sets the backdrop for the research of this thesis.

Developments in the past decade in compact, high speed processing units, have opened the door to development of more sophisticated, independent and more capable systems than ever before.

"The process of automating attitude determination and control activity will allow systems which are more complex and more responsive. In general, attitude control systems will go from being complex analog electronics to being mechanically simpler, but logically far more complex, digital systems, driven principally by software" (Wertz, 2001: 174). This increased sophistication and capability will allow for performance increases for satellites, which brings the focus back to the space based laser, or similar systems. Specifically, it opens the door for these satellites to carry out wide angle slewing maneuvers reliably, and in less time than what is required by most of today's satellite control systems.

Objectives

There were several objectives for this thesis. The first objective was to design a satellite attitude control algorithm to provide near-optimal control to a satellite undergoing a large, multi-axis slewing maneuver, given full state feedback. This objective has practical implications for reasons given above—the next generation of satellites may require it.

The second objective was to implement this controller on SIMSAT, AFIT's simulated satellite system (Figure 3), using SIMSAT's control wheels as the only control input. This would serve a dual purpose. First, the experimental verification of the first objective adds confidence to the computer modeling performed to produce the controller. Second, it also provides a teaching tool for AFIT. Students trained at AFIT can get a

better understanding of attitude control given demonstrations on SIMSAT, which did not have a working controller for wide angle maneuvers.

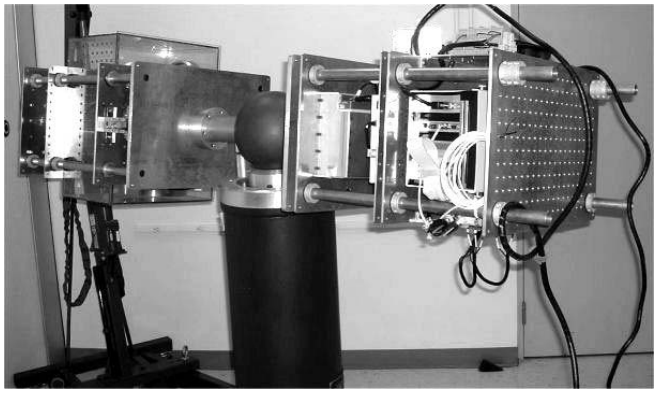


Figure 3. SIMSAT

The third objective was to design a thruster system and install it on SIMSAT. There were several reasons for this objective. One was to realize the fullest extent of the maneuvers the controller is capable of performing, without encountering the saturation problems inherent with a reaction wheel system. Second was to allow for redundant controls of SIMSAT and expand its teaching potential. Finally, it allows the study of hybrid control techniques.

The final objective was to design and implement a hybrid controller. The intent was to pass control between the reaction wheels and thrusters in a logical fashion, taking advantages of the strengths of each control method. This would allow SIMSAT to undergo large-angle slewing maneuvers, while at the same time allowing it a higher degree of pointing accuracy than gas jets alone would provide.

Methodology

The first step in the development and testing of the control algorithms was to create a computer model of SIMSAT, to be programmed in Simulink. After the satellite was modeled, the controller had to be programmed. First, a simple proportional-differential controller was built, then a linear quadratic regulator (LQR) powered State Dependent Riccati Equation (SDRE) controller was built.

After the controller was proven to operate successfully on the computer model, it needed to be tested experimentally. The Air Force Institute of Technology owns and operates one of the few three degree-of-freedom satellite simulators in the country—SIMSAT. SIMSAT provides a convenient tool for testing satellite control algorithms, using MATLAB, Simulink, dSpace® and Real Time Workshop software. The controller developed for the computer model of SIMSAT could be directly implemented on SIMSAT and tested.

Thruster construction was the next objective. A thruster had to be designed that would be simple, lightweight and powerful enough to achieve the desired performance. This thruster then would have to be installed on SIMSAT and connected to its control architecture. It also needed to be safe for use in the lab.

The final stage was the hybrid controller design. At this stage, the controller had to be redesigned to switch between thruster control and reaction wheels to maximize performance. An "energy" based system was chosen for this switching mechanism.

II. Literature Review

Present Day Control Techniques

Several methods of control for 3-axis-stabilized satellite maneuvers exist. The most common types used on satellites are proportional-differential (PD) and proportional-integral (PI) controllers.

> Large angle attitude manoeuvres of three-axis stabilized spacecraft are currently usually performed by means of a sequence of single axis rotations. Such manoeuvres may take a relatively long time and their duration can often be considerably reduced by performing simultaneous three-axis or slew manoeuvres. These manoeuvres are necessarily more complex but can be effected with moderate on-board computer power (Dempster, 1983: 113).

With the simultaneous rise of computational power and need for more agile space systems, more capable control techniques have been developed.

Non-linear control algorithms

Several types of non-linear controllers are presented in the literature. Those techniques include Lyapunov stability techniques, two-point boundary value problems (TPBVP) and State Dependent Riccati Equations (SDRE).

Lyapunov Stability Techniques

The following is a description of Lyapunov stability.

> The word *stable* (derived from the Latin adjective *stabilem,* to stand firmly) has many shades of meaning and therefore must be precisely defined if it is to be of use in the study of spacecraft attitude dynamics and control. Two such stability concepts are related to linear and nonlinear systems. The first concept is that of "infinitessimal" or Lagrange stability, which is a "boundedness" concept applicable to linear systems. If, for example, a small deviation from some equilibrium point remains bounded, then the motion is said to be Lagrange, or infinitessimally stable. This is, in general, applicable to systems which can be described by linear differential equations, the equilibria or the solutions of which are either stable or asymptotically stable when the perturbed motion approaches the equilibrium condition at some time in the future.

> The second stability concept requires that a solution to a differential equation, beginning sufficiently close to the equilibrium (origin), must remain arbitrarily close to itself after a perturbation. This concept…is applicable to the solutions described by both linear and nonlinear differential equations.
>
> Lagrange stability thus required only that the solution (trajectory) remain within a finite distance from equilibrium while Liapunov stability requires arbitrarily small deviations after a perturbation from equilibrium. Therefore, Liapunov stability implies Lagrange stability, but the opposite is not true (Chobotov, 1991: 113).

The controller described in "Investigation of the Stability of Satellite Large Angle Attitude Manoeuvres Using Nonlinear Optimization Methods" uses a controller based on Lyapunov Stability Theory.

> The stability of three-axis attitude manoeuvres has been investigated quite some time ago…using Lyapunov stability theory. These investigations have taken into account the cross coupling of spacecraft axes during slew manoeuvres, but they have ignored the nonlinear constraints imposed on the controlled dynamical system by available attitude control actuators such as gas jets and reaction wheels. The present paper extends the above results by taking into account the "hard" nonlinear characteristics of attitude control actuators. Numerical simulations indicate the global (asymptotic) stability of the resulting more realistic system and numerical experiments are described with a domain of attraction (DOA) estimation procedure based on nested nonlinear optimization routines which could potentially prove the required system stability (Dempster, 1983: 113).

The paper goes on to give examples of test cases that prove global asymptotic stability for the controller.

Two-point boundary value problems

The two-point boundary value problem (TPBVP) occurs when both the initial and final time are known, and some optimization of what occurs in between takes place. An effort of this type was presented in a dissertation by a doctoral student at Virginia Tech. It offered "optimal reaction wheel control…capable of achieving specific attitude and angular velocity states at a specified final time while minimizing the integral over time of

the power squared subject to arbitrary time-varying limits on the three torques" (Skaar, 1982: 65).

The purpose of the time varying torque limits were given.

a) to limit the largest torque required during the maneuver.
b) to avoid discontinuous jumps in the control at the initial and final times.
c) to allow for specifiable torque-up and torque-down control histories suited to hardware capabilities and/or spacecraft elastic properties.

The concentration on minimizing torque requirements is of note, due to a similar concern with propellant usage; however, the use of a two-point boundary value problem does not offer the flexibility desired for this application, as the times are fixed and the performance goals are in terms of the torque used. Additionally, it is uncertain that the dynamics of this thesis could be adequately represented due to the numerous non-linearities involved.

State Dependent Riccati Equation

While no papers were presented on the direct application of State Dependent Riccati Equation (SDRE) on spacecraft wide angle slewing, there were several that addressed it as a means of control. One AFIT thesis used the technique to control reconfiguration of satellite formations (Irvin, 2001). This thesis work was followed up by more work and similar application of the SDRE approach (Carraher, 2002).

The SDRE solution offers advantages to both the Lyapunov Stability approach and TPBVPs. First, it offers flexibility. It is a free final time method, adaptive to unexpected changes in system dynamics. Second, it is simpler. In effect, the routine takes advantage of modern computing power, solving the linear quadratic regulator (LQR) problem for each time step, or linearizing the solution every 0.05 seconds—the

time step used for all models and implemented controllers in this thesis. For these reasons, it was chosen as the type of control to be pursued.

Precision Pointing

Several texts gave general guidance and specific examples of pointing accuracy depending on the control method. Below is listed typical accuracies of several pointing devices (Table 2).

Method	Typical Accuracy	Remarks
Spin Stabilized	0.1 deg	Passive, simple; single axis inertial, low cost, need slip rings
Gravity Gradient	1-3 deg	Passive, simple; central body oriented; low cost
Jets	0.1 deg	Consumables required, fast; high cost
Magnetic	1 deg	Near Earth; slow ; low weight, low cost
Reaction Wheels	0.01 deg	Internal torque; requires other momentum control; high power, cost

3-axis stabilized, active control most common choice for precision spacecraft

Table 2. Control method vs. typical pointing accuracy (de Weck, 2001).

Notice the reaction wheels allow the greatest accuracy. This table of "typical values" does not preclude increased accuracy to values given. In fact, the Hubble Space Telescope (HST), an example cited in Chapter 1, has a pointing accuracy of .007 arcseconds, provided by reaction wheels, far exceeding the value given (Jane's, 2002: 453). Reaction wheels capable of maintaining such precise pointing do not produce a

large slew rate. For the case of the HST, this is not a large concern, but for applications which need pointing accuracy and a high slew rate, hybrid control will be required.

Similar Efforts

SADSaC

A single degree of freedom satellite simulator called "SADSaC," for "Satellite Attitude Dynamics Simulation and Control," was built by a student at the University of Colorado, Colorado Springs, using both wheels and gas jets for single-axis control (St. Pierre, 1998). In this physical simulator, a proportional-integral controller was used for the controller. This simulator also provided full-state feedback. The thrusters and control wheels were controlled separately and no mention is made of hybrid control in the project. The pointing accuracy goal for the project was 0.5 deg. A pointing accuracy of 0.4 deg was attained using reaction wheel control after compensating for drift in the rate sensing system and a physical imbalance (St. Pierre, 1998: 69). For the gas jet system, acting independently, the best pointing accuracy achieved was 0.9 deg. The SADSaC also reported a problem with "chatter caused in the hysteresis band of the control circuit."

III. Design

Overview

There were three basic design goals for this thesis. The first two involved software design and the third, a physical design. First, a computer model of SIMSAT had to be programmed. The computer model for SIMSAT went through several evolutions. It began as a basic simulation of SIMSAT, with many properties estimated. As it became possible to substitute experimentally obtained values in for the estimates, the model began to produce a more accurate representation of SIMSAT's actual behavior. Second, two PD and SDRE controllers had to be designed. These controllers had to be implemented on the computer model and then on the physical SIMSAT. Finally, a thruster system had to be designed. This system would be tested as an individual controller, and then in tandem with the reaction wheels.

SIMSAT

Before discussing the computer model or the experimental setup, it is necessary for the reader to understand exactly what SIMSAT is. SIMSAT was designed and constructed in 1999 by Captains Colebank, Jones, Nagy, Pollak and Lt Mannebach as part of their masters thesis work to serve as a satellite system simulator and experimental test bed for Air Force related research topics. SIMSAT is a sophisticated system, designed to simulate a zero torque environment, while providing full-state feedback in terms of angular positions and rates of change and control input states. For a detailed description of the original design, refer to the thesis document (Colebank, 1999).

Since it was constructed, SIMSAT has evolved, as it did for work associated with this thesis. For the present purpose, SIMSAT can be described as it was before its

modification for this thesis in three basic parts: physical properties, interface, and control input.

Physical Properties

SIMSAT is approximately 250 lbs and 72 x 21 x 14 inches in size (Figure 4). It is supported on an air bearing by a micro thin layer of air, which supports its 8-inch diameter spherical centerpiece. From the spherical centerpiece extend two arms. Each arm supports a rack assembly. On one rack was mounted the batteries, providing bus voltages of 12V, 24V and 36V, a 3-axis gyro, an onboard computer and I/O board. On the other was mounted three independent reaction wheels. This is the side that was chosen for thruster installation as well, due mostly to available space on that side for mounting the thruster assembly. This will be discussed later (see Chapter 3, Thruster Installation).

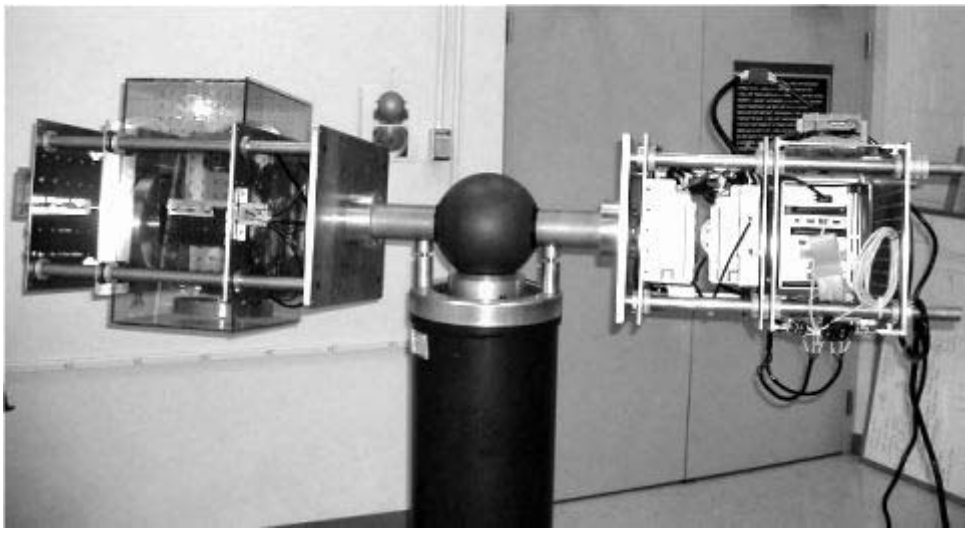


Figure 4. SIMSAT prior to modifications

SIMSAT was designed to be rigid, and therefore keep its center of gravity at or very near to the center of the central sphere. However, this is not the case. SIMSAT

actually sags to an equilibrium position, regardless of any attempt to balance it. This obstructs attempts to rotate it to other stable positions about either the pitch or the roll axis, the latter more severely. If one attempts to roll or pitch SIMSAT to an arbitrary position, it will seek its equilibrium position upon control release. This flexibility precludes true three-axis control of SIMSAT.

Body frame spin rates are provided via a gyro. With integration of these rates, this gyro allows feedback of both velocity and position information. However, the gyro is subject to large amounts of noise and, therefore, drift. The pitch axis (b_2) axis is affected worst. It is not uncommon for the pitch axis gyro to drift approximately 20 degrees in 60 seconds. This limits the testing capability in the pitch axis.

Interface

SIMSAT is equipped with a dSpace Autobox® 1005 computer. Installed in the computer are three input/output (I/O) cards, a ds2003, ds2103 and ds4201S. The Autobox is connected to the "ground station" lab computer via a RadioLAN® Wireless DockLINK Model 408 wireless local area network. The interface from the lab computer to the Autobox® is the software program dSpace Control Desk®. This program is capable of directly implementing control schemes built in Matlab® Simulink, which provided a graphic user interface facilitating construction of both the controllers for, and the computer simulation of, SIMSAT.

Control

SIMSAT originally used three reaction wheels (Figure 5) for control input. The wheels, mounted on three orthogonal axes, are 8 ½ inches in diameter and 4 ½ pound per wheel. They are accelerated using Animatics Smart Motor 3450. These motors are

capable of being tuned in several ways, including an acceleration rate limit factor. This acceleration factor was reduced for the purposes of this thesis, to allow the wheels to be used for fine pointing. For further discussion of motor tuning and commanding, refer to the masters thesis by Dabrowski in the References.

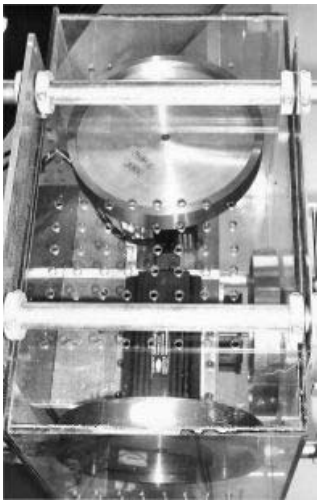


Figure 5. SIMSAT reaction wheels.

Computer Simulation of SIMSAT

For the computer model of SIMSAT, the position of SIMSAT needed to be described for its three degrees of freedom (rotation about three orthogonal body axes), so frames of reference had to be established. First, the inertial reference frame will be briefly introduced, then the satellite body frame.

Reference Frames

The inertial reference frame establishes a guide for observations of SIMSAT. A logical frame for SIMSAT is earth-fixed. This frame is described using three axes, i_1, i_2 and i_3, to represent the positive orthogonal unit vectors of the frame. The i_3 axis points up in the local vertical. The i_1 and i_2 vectors are perpendicular to this axis and each other,

forming an orthogonal, right-handed coordinate set. The orientation of this set about the i_3 axis is arbitrary, thus it will be assumed that the inertial frame will be aligned with the body frame, described below, at the beginning of each experiment.

The body frame is described by the unit vectors b_1, b_2 and b_3 (Figure 6). Physically matching SIMSAT to the body frame, the b_1 axis was chosen to be its long axis.

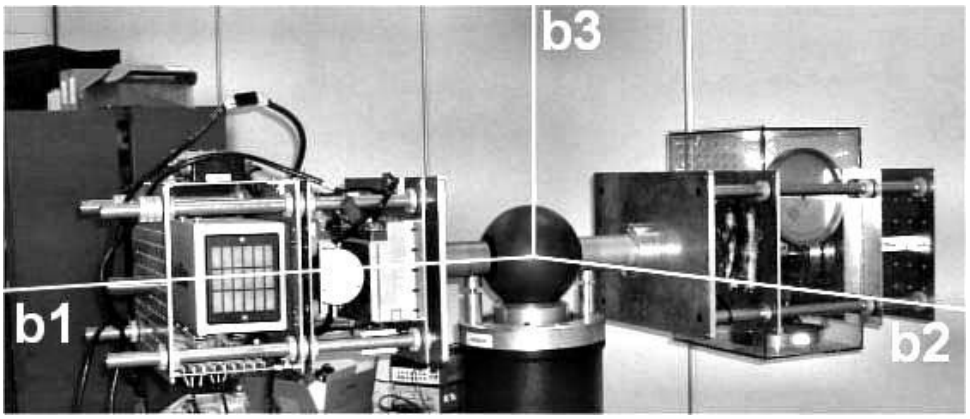


Figure 6. SIMSAT shown with body-fixed axes.

To properly orient the b_2 axis, the reader must understand that the side with the wireless LAN antenna was considered the "top" of SIMSAT (Figure 7).

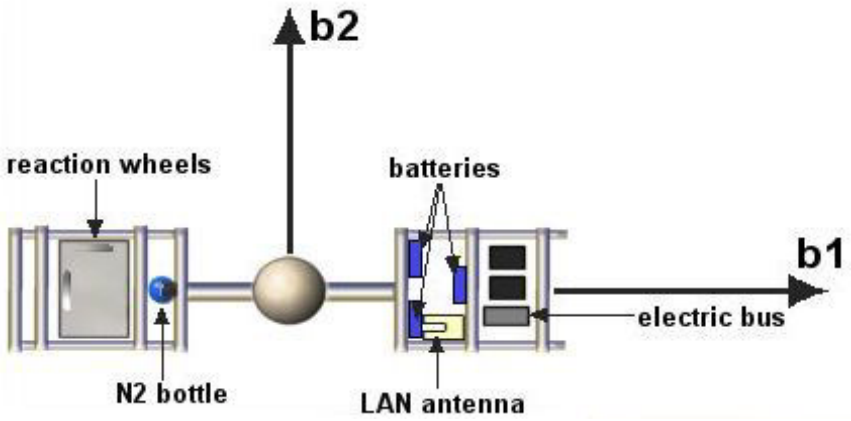


Figure 7. Body frame points of reference (top view).

The side, perpendicular with the top, away from the power bus, was considered the "front" of SIMSAT. The b_2 axis was chosen to be an axis perpendicular to the long (b_1) axis that extended from the center of the sphere toward the front of SIMSAT. The b_3 axis extends from the center of the sphere and extends toward the top plane, perpendicular to it. These three axes constitute a right-handed orthogonal set.

The designers of SIMSAT chose to represent its attitude using Euler 3-2-1 rotation angles (Figure 8). The orientation of the satellite is described, compared to an inertially fixed starting position, by a first angle, θ_1, the angle of the first rotation about the satellites b_3 axis. From this orientation, the satellite is rotated by an angle of θ_2 about its b_2 axis. Finally, it is rotated θ_3 degrees about its b_1 axis. These three rotations lead to a unique orientation of SIMSAT within the inertial frame.

It should be noted that the selection of 3-2-1 Euler angles was intentional, as it makes it impossible for this sequence's singularity, θ_2=90 deg, to be reached. SIMSAT is incapable of achieving this attitude due to the location of the supporting air-bearing post.

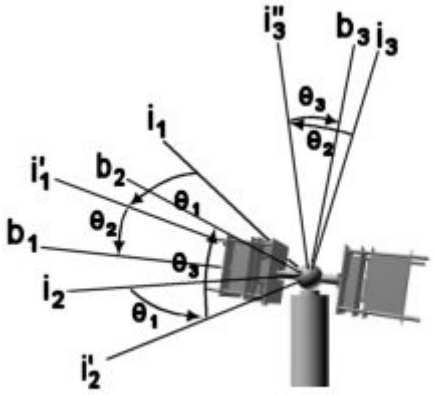


Figure 8. 3-2-1 rotation sequence.

Due to this post, the designers of SIMSAT recognized that its range of motion would be restricted. The approximate maximum angle θ_2 can reach is 30 degrees. Both θ_1 and θ_3 are unlimited in angular motion.

To translate data between the body and inertial frames, it is necessary to identify the rotation matrix for the satellite. It was derived, beginning with the single axis rotations described above, and is given below as Equation 1 (Wiesel, 1997: 114). The rotation matrix given, if multiplied by the inertial frame coordinated, will give the body frame coordinates.

$$R_{3b} = \begin{bmatrix} \cos(\theta_1)\cos(\theta_2) & \sin(\theta_1)\cos(\theta_2) & -\sin(\theta_2) \\ \cos(\theta_1)\sin(\theta_2)\sin(\theta_3) - \sin(\theta_1)\cos(\theta_3) & \cos(\theta_1)\cos(\theta_3) + \sin(\theta_1)\sin(\theta_2)\sin(\theta_3) & \cos(\theta_2)\sin(\theta_3) \\ \cos(\theta_1)\sin(\theta_2)\cos(\theta_3) + \sin(\theta_1)\sin(\theta_3) & \sin(\theta_1)\sin(\theta_2)\cos(\theta_3) - \cos(\theta_1)\sin(\theta_3) & \cos(\theta_2)\cos(\theta_3) \end{bmatrix} \quad (1)$$

Next, it is necessary to describe how these angles change with time. Start with the body frame rates of change with respect to the Euler angles and derivatives (Wiesel, 1997: 115).

$$\omega_1 = \theta'_3 - \theta'_1 \cdot \sin(\theta_2) \quad (2)$$

$$\omega_2 = \theta'_2 \cdot \cos(\theta_3) + \theta'_1 \cdot \cos(\theta_2) \cdot \sin(\theta_3) \quad (3)$$

$$\omega_3 = -\theta'_2 \cdot \sin(\theta_3) + \theta'_1 \cdot \cos(\theta_2) \cdot \cos(\theta_3) \quad (4)$$

From these, the Euler rates can be obtained by solving for the above equations simultaneously.

$$\theta'_1 = \frac{(\sin(\theta_3) \cdot \omega_2 + \omega_3 \cdot \cos(\theta_3))}{\cos(\theta_2)} \quad (5)$$

$$\theta'_2 = \omega_2 \cdot \cos(\theta_3) - \sin(\theta_3) \cdot \omega_3 \quad (6)$$

$$\theta'_3 = \frac{(\sin(\theta_3)\cdot\omega_2\cdot\sin(\theta_2) + \cos(\theta_2)\cdot\omega_1 + \omega_3\cdot\sin(\theta_2)\cdot\cos(\theta_3))}{\cos(\theta_2)} \tag{7}$$

This completes the first building block of the computer model. Given body angular rates and instantaneous measurements of Euler angles, Euler rates can be calculated. By integrating these Euler rates with respect to time, subsequent Euler angles, due to current rates and the time step, can be computed.

With these relationships, the orientation can be described. The 3-2-1 Euler angles describe the relationship between the inertial frame and the current body fixed reference frame. Additionally, the rates of change of these angles can be determined from the angles themselves and the body fixed angular rates as can be seen in Equations 5-7.

Plant Modeling

The next task is to compute the body angular rates. These rates can be produced from a set of equations describing the angular dynamics of a rigid body (Wiesel, 1997: 111).

$$M_1 = I_{11}\cdot\omega'_1 + (I_{33} - I_{22})\cdot\omega_2\cdot\omega_3 \tag{8}$$

$$M_2 = I_{22}\cdot\omega'_2 + (I_{11} - I_{33})\cdot\omega_1\cdot\omega_3 \tag{9}$$

$$M_3 = I_{33}\cdot\omega'_3 + (I_{22} - I_{11})\cdot\omega_1\cdot\omega_2 \tag{10}$$

where M_1 is an applied external moment about the b_1 axis, M_2 is about the b_2 axis and M_3 is about the b_3 axis. Each I_{ii} is the mass moment of inertia about the i^{th} principle axis of the body frame, coinciding with the above-established b-frame. The symbol ω_i is the angular velocity about the b_i axis, and ω'_i is the rate of change of that angular velocity.

With that, the system has been mathematically described with the inputs M_i, which influence the attitude of the system.

The orientation may be changed using the reaction wheels. To examine how, begin with the equation for the sum of a moment about an axis with no external torques.

$$\Sigma M_i = 0 \tag{11}$$

From there, wheel acceleration effects on SIMSAT can be obtained by modeling them as external torques.

$$M_i = -I_w \cdot \left(\Omega'_i + \omega'_i\right) \tag{12}$$

where i is any of the three axes, I_w is the mass moment of inertia of the three reaction wheels (all are equal), Ω_i is the angular velocity of that wheel, and Ω'_i is the angular acceleration of that wheel. Matching Equation 12 with Equations 8-10 and rearranging, one gets equations for SIMSAT angular acceleration as a function of SIMSAT angular rates and reaction wheel acceleration.

$$\omega'_1 = \frac{\left(I_{22} - I_{33}\right)\cdot\omega_2\cdot\omega_3 - I_w\cdot\Omega'_1}{I_{11} + I_w} \tag{13}$$

$$\omega'_2 = \frac{\left(I_{33} - I_{11}\right)\cdot\omega_1\cdot\omega_3 - I_w\cdot\Omega'_2}{I_{22} + I_w} \tag{14}$$

$$\omega'_3 = \frac{\left(I_{11} - I_{22}\right)\cdot\omega_1\cdot\omega_2 - I_w\cdot\Omega'_3}{I_{33} + I_w} \tag{15}$$

The rates of change of SIMSAT angular velocities with time has been obtained, depending only on the reaction wheels. Its angular velocities can be obtained by integration.

It is now necessary to describe the thrusters mathematically. In Equation 12, the torques caused by the reaction wheels were modeled as external. The thrusters will be treated likewise.

$$M_i = T \cdot d_i \tag{16}$$

where M_i is the moment about axis i, T is the thrust (for now, presuming all thrusters produce the same thrust), and d_i is the distance between the thrust axis and the CG of SIMSAT.

With this new external torque, Equation 12 can be rewritten to include thrusters.

$$M_i = -I_w \cdot \left(\Omega'_i + \omega'_i\right) + T \cdot d_i \tag{17}$$

and Equations 13-15 become

$$\omega'_1 = \frac{\left(I_{22} - I_{33}\right) \cdot \omega_2 \cdot \omega_3 - I_w \cdot \Omega'_1 + T_1 \cdot d_1}{I_{11} + I_w} \tag{18}$$

$$\omega'_2 = \frac{\left(I_{33} - I_{11}\right) \cdot \omega_1 \cdot \omega_3 - I_w \cdot \Omega'_2 + T_2 \cdot d_2}{I_{22} + I_w} \tag{19}$$

$$\omega'_3 = \frac{\left(I_{11} - I_{22}\right) \cdot \omega_1 \cdot \omega_2 - I_w \cdot \Omega'_3 + T_3 \cdot d_3}{I_{33} + I_w} \tag{20}$$

Equations 5-7 and 18-20 formed the main part of SIMSAT's dynamics and were programmed into Simulink to form the plant computer model (Figure 9). For more detail of the Simulink computer model, refer to Appendix A.

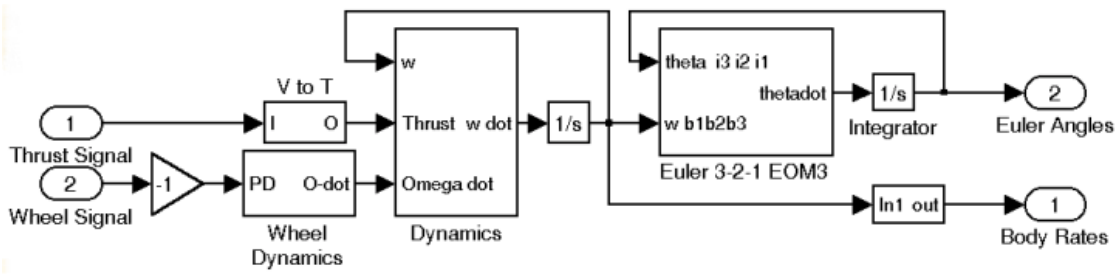


Figure 9. Plant Model

The dynamics of how SIMSAT responds to command inputs of thrust or reaction wheel acceleration have been discussed. It still remains, however, to describe how both the reaction wheels and the thrusters respond to command inputs.

Next, a discussion of how the modeled reaction wheels respond to control inputs is required. The control input for the wheels is the change in reaction wheel speed, $\delta\Omega$. The physical limitations of the motor that drives the wheel does not allow it to provide these changes instantaneously. The wheel acceleration responds in a proportional manner for a small enough request, but beyond some fixed limit, it begins to saturate. Simulink allows saturation of this type to be modeled through the use of look-up tables (Figure 10).

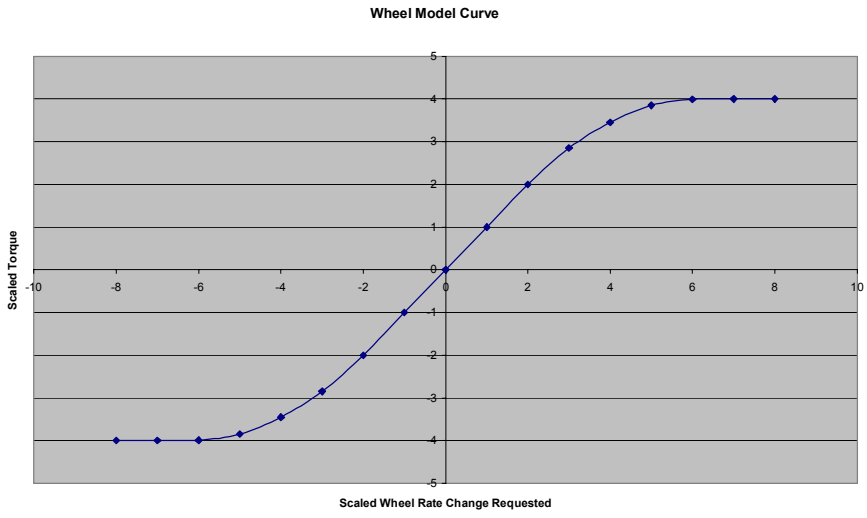


Figure 10. Scalable reaction wheel dynamics.

This chart is built from the data used in the wheel acceleration look-up table. It is scaled by both an input gain, $K_{motor,in}$, and an output gain, K_{motor}, to assist in model matching.

Finally, a discussion on how the thruster system responds to command inputs is required. The thruster system receives the command signal and compares that signal to a dead band limit set by the initializing software. If the signal surpasses this dead band limit on the positive side, the positive thrusters fire. Likewise, if a negative command surpasses the magnitude of the dead band constant, the negative thrusters fire.

Control

Two types of controllers were built to manage the task of generating control inputs. The first was a proportional-differential (PD) controller. The second was a linear quadratic regulator (LQR) based, State Dependent Riccati Equation (SDRE) controller.

PD Control.

Proportional-derivative (PD) control allows control to be attained through the assignment of gains to the error signal and the state derivative (Figure 11).

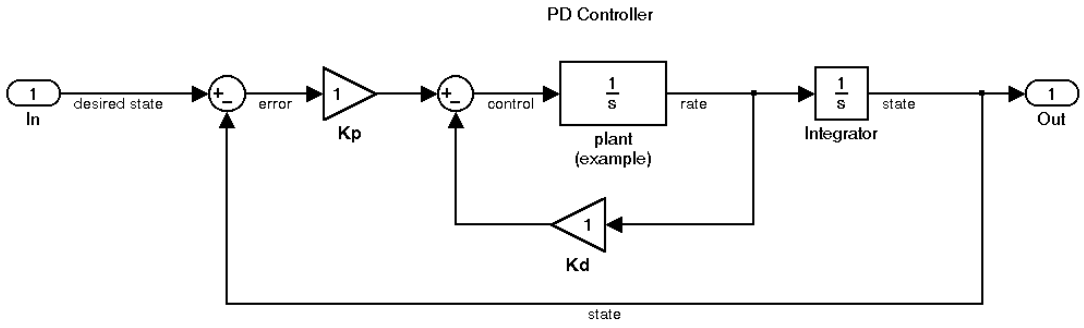


Figure 11. Example PD Controller

A desired state is introduced into a system. The actual state is subtracted from the desired state to produce the error signal. This error signal is multiplied by the proportional gain, K_p. From this product is subtracted the state rate of change multiplied by the derivative gain, K_d. The resulting signal is the control signal. This signal affects the plant according to the dynamics of the plant, and the rate of change of the state is observed. This rate of change, as stated before, is fed back for forming the control signal, and is also integrated to produce the current state. The state is also fed back to form the control signal and is observed.

In the case of this thesis, the state is the vector of Euler angles and the rate is the vector of Euler angle rates of change, given in Equations 5-7. The difficulty in implementing a PD controller arises due to several non-linearities within SIMSAT: reaction wheel control dynamics, bang-bang thrust control and coupled kinematics equations.

The reaction wheels do not respond in a linear fashion to command inputs. The precise response curve to an input is unknown, but it is clear that the wheels become saturated on two levels. First, they have a maximum rate of acceleration. Second, they have a maximum speed.

On-off thrust control is another non-linearity. The thrusters are set to a specific output pressure and allowed an approximately constant flow rate, resulting in a constant thrust. This is in conflict with classical PD control, which assumes proportional control.

Another non-linearity arises when multiple-axis control is desired. When only singe-axis control is desired, an Euler angle displacement can be reduced by implementing control about the corresponding body-fixed axis. For example, consider

the case where the satellite is at rest with all Euler angles equal to zero. If the desired orientation of the satellite is $\theta_1 = 120$ degrees, control will be implemented about the b_3 axis to achieve that orientation successfully. When multiple-axis control is desired, however, this is not the case. For example, if the desired orientation is $\theta_1 = 120$ degrees and $\theta_3 = 120$ degrees, a problem arises. When SIMSAT begins to roll, a component of the control about the b_3 axis attempting to reduce $\Delta\theta_1$ will begin to affect $\Delta\theta_2$, which then must be counteracted by the control about the b_2 axis, which in turn begins to affect a component of its control about the yaw axis. As θ_3 increases, this effect grows. Finally, once SIMSAT has rolled past $\theta_3 = 90$ degrees, control actions about the b_3 axis change signs, making the system unstable.

Finally, as seen in Equations 18-20, the kinematics of SIMSAT are coupled. This factor can become significant for rapid three-axis motion. However, for single-axis maneuvers, it does not pose a problem.

SDRE Control.

The second type of controller designed for application was the LQR/SDRE controller. This type of control requires a state space representation. First, the state to be controlled must to be defined.

$$x = \begin{bmatrix} \theta_1 \\ \theta_2 \\ \theta_3 \\ \omega_1 \\ \omega_2 \\ \omega_3 \end{bmatrix} \qquad (21)$$

The control vector u must also be defined. For the case of this thesis, the vector would be composed of a vector of reaction wheel accelerations or thrust.

$$u = \begin{vmatrix} \Omega_1' \\ \Omega_2' \\ \Omega_3' \end{vmatrix} \tag{22}$$

$$u = \begin{vmatrix} T_1 \\ T_2 \\ T_3 \end{vmatrix} \tag{23}$$

Next, because LQR and SDRE use state space representations, the A, B, C and D matrices must be defined. The general representation for the state space matrices is given by

$$\frac{dx}{dt} = A \cdot x + B \cdot u \tag{24}$$

$$y = C \cdot x + D \cdot u \tag{25}$$

where A is derived from the dynamics of the system being described, B describes how the control inputs affect the states, C determines what is being observed, and D allows control inputs to be observed directly.

Given the state space representation, the goal is to determine a gain vector, K, that when multiplied by the state vector, is fed back to determine the control input to be applied.

$$u(t) = -K \cdot x(t) \tag{26}$$

Now, the concept of Linear Quadratic Regulator (LQR) control must be introduced. For this type of control, a performance index, J, is introduced (Equation 27).

$$J = \frac{1}{2}\cdot\int_0^{\infty}\left(x^T Q\cdot x + u^T R\cdot u\right) dt \tag{27}$$

LQR control minimizes this performance index J, given the weighting matrices Q and R, and provides the gain described above. A more in depth discussion of LQR control is presented in Modern Control Systems by Ogata (Ogata, 2002: 897). The goal is to obtain K_{lqr}, the steady state gain matrix resulting from the solution of the Algebraic Riccati Equation (ARE). This gain is multiplied by the state vector to compute required control.

Before continuing, it is important to distinguish the difference between LQR control and State Dependent Riccati Equation (SDRE) control. LQR is applied to systems that have a constant A matrix. That is, the system's dynamics do not change with time. The dynamics of SIMSAT, as seen in Equations 13-15, are coupled. To use the LQR approach, the system is effectively linearized by applying current time step values to the A matrix. Because the results this linearization degrade as the linearization point is departed from, the A matrix must be updated at every time step, as the state changes.

The A matrix for SIMSAT is derived directly from Equations 5-7 and parts of 10-12, and will be dependent on the state, in other words, not constant. It has six rows and six columns, each equal to the number of states.

$$A = \begin{bmatrix} 0\ 0\ 0 & 0 & \frac{-\sin(\theta_3)}{\cos(\theta_2)} & \frac{-\cos(\theta_3)}{\cos(\theta_2)} \\ 0\ 0\ 0 & 0 & -\cos(\theta_3) & \sin(\theta_3) \\ 0\ 0\ 0 & -1 & \frac{-\sin(\theta_3)\cdot\sin(\theta_2)}{\cos(\theta_2)} & \frac{-\sin(\theta_2)\cdot\cos(\theta_3)}{\cos(\theta_2)} \\ 0\ 0\ 0 & 0 & \frac{(B-C)\cdot\omega_3}{A+I_W} & 0 \\ 0\ 0\ 0 & 0 & 0 & \frac{(C-A)\cdot\omega_1}{B+I_W} \\ 0\ 0\ 0 & \frac{(A-B)\cdot\omega_2}{C+I_W} & 0 & 0 \end{bmatrix} \quad (28)$$

In this representation and the representation of the B matrix below, A, B and C have been substituted for the MOIs I_{11}, I_{22} and I_{33}, respectively. As stated before, A is dependent on the state vector; therefore, in the simulation, A must be solved for at every time step.

B can be derived using the remaining portion of Equations 13-15, those not dependent on the A-matrix above. These apply to the control input. The B-matrix will have as many rows as the number of states (six in this case), and as many columns as there are control inputs (three).

$$B = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ \frac{-I_W}{A+I_W} & 0 & 0 \\ 0 & \frac{-I_W}{B+I_W} & 0 \\ 0 & 0 & \frac{-I_W}{C+I_W} \end{bmatrix} \quad (29)$$

A similar matrix can be derived for thrusters.

Since the system provides full state feedback, C, the observability matrix, will just be the identity matrix.

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (30)$$

And because there is no feed-forward of our the inputs, the D-matrix is all zeros.

$$D = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (31)$$

That leaves only the two weighting matrices, Q and R, from the cost function given in equation 27 to be discussed and defined. The diagonal of Q is the penalty placed on the controller for not achieving desired state. For example, Q_{11} (row 1, column 1 of the matrix Q) is the penalty for not achieving state $\Delta\theta_1=0$, Q_{22} is the penalty for not achieving state $\Delta\theta_2=0$ and so on. The diagonal of the matrix R is the penalty for using the associated control input. Therefore R_{11} is the penalty for non-zero Ω_1, R_{22} is the penalty for non-zero Ω_2, and R_{33} is the penalty for non-zero Ω_3. An example of Q and R where all state and controls are equally penalized is shown below in Equations 32 and 33.

$$Q = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad (32)$$

$$R = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (33)$$

The general formulation for the Q and R matrices allows cross-weighting of states. For the purpose of this thesis, the off-diagonal terms for both Q and R will always be equal to zero.

Again, the output of concern from the LQR routine is a gain K_{lqr}, which can be obtained from MATLAB® using the "LQR" command, and which can be multiplied by the present state to derive the control input desired at the present time.

Using the SDRE control technique, it was possible to orient SIMSAT computer model to any desired orientation. The results of the SDRE control on SIMSAT are contained in Chapter 4. To see the Simulink model of the SDRE controlled computer SIMSAT model, see Appendix A. For launching MATLAB® code for the model, see Appendix B.

Control Switching.

For hybrid control, a key aspect is when to transition control between thrusters and reaction wheels. Both the PD and SDRE controllers used an "energy switch" to perform this function. This switch activated based on the energy state of the system.

The overall energy of the system was defined as the sum of "potential" energy, V, and the "kinetic" energy, T. This provides a convenient method of determining when SIMSAT is both near its target attitude and at a low angular velocity.

$$E = V + T \qquad (34)$$

These are not the classical definitions of potential and kinetic energy. For the purposes of the switch, the potential energy was defined as a scaling constant multiplied by the position error squared.

$$V = k_V \cdot \Delta\theta^2 \qquad (35)$$

The kinetic energy was defined as a scaling constant multiplied by the square of the angular rate.

$$T = k_T \cdot \omega^2 \qquad (36)$$

In the last two equations, k_V and k_T are scaling constants, chosen to properly shape and scale the energy curve (Figure 12). This figure shows the angular position at the top, the angular rate in the middle and the energy on the lowest chart. Comparing the equations above to Figure 12, it is clear that the position error is scaled to weigh heavier in the calculation of energy. The scaling of the kinetic and potential energies was determined by trial and error, and the tradeoffs in changing the scaling were not thoroughly investigated and remain as a topic for future study.

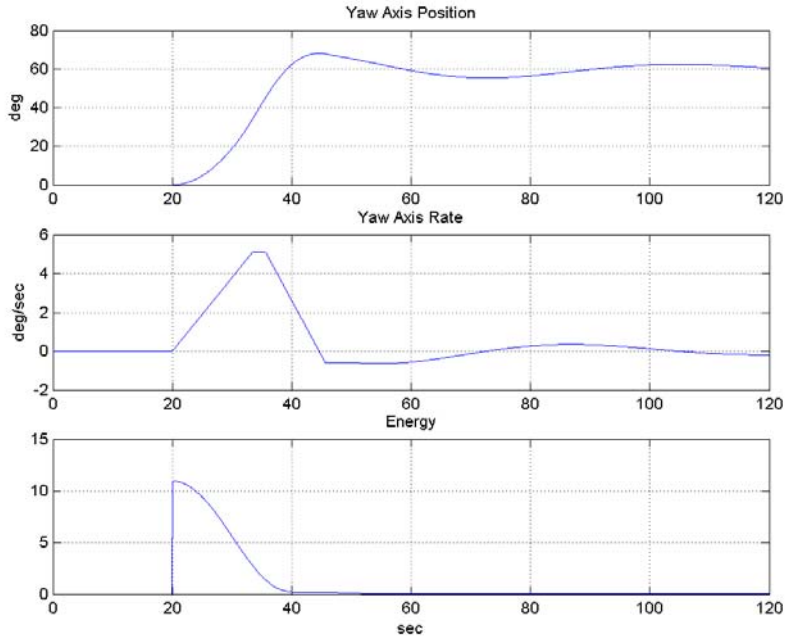


Figure 12. Energy curve and components for single-axis maneuver

By adding V and T, properly scaled, a good representation of the overall angular energy required of the control inputs was readily available. At a predetermined threshold, the controller would switch between the two control inputs.

The question of where to set the threshold requires discussion. There are several competing interests involved. First, it is desirable to conserve thruster fuel. However, it is also desirable to achieve a low settling time. If the threshold for control transfer is set too low, excessive thruster fuel will be expended. If it is set too high, the settling time will be longer. A more in-depth discussion of threshold selection is presented in Chapter 4, Hybrid Control.

Thruster Installation

Another part of this thesis was to modify SIMSAT to allow for a second type of control—thrusters. The thrusters had to meet several requirements. First, they would

have to provide sufficient torque to allow rapid, large-angle slewing maneuvers, a capability that SIMSAT did not previously have due to reaction wheel saturation. Second, the system would have to be safe for the lab environment. Third, it had to be constructed at low cost using off-the-shelf products. Finally, the thruster system would have to be installable on SIMSAT, which meant it had to be relatively small, lightweight, and had to be powered via SIMSAT's array of batteries.

With all of these considerations in mind, the design chosen was based on high-pressure gaseous nitrogen. The source of the nitrogen would be an 88 cubic inch nitrogen gas bottle, with a maximum operating pressure of 4500 PSI. The gas bottle chosen for this was the PMI Pure Energy 88 cubic inch 4500 psi Fixed Nitrogen System (Figure 13), a bottle manufactured for use in paintball. It was found that the manufacturers of these bottles had developed bottles that would provide maximum "thrust" for minimum weight and size, while at the same time considering safety implications (the manufacturer has never had a reported case of a bottle rupturing).



Figure 13. PMI® Pure Energy 88 cubic inch, 4500 psi Fixed Nitrogen System

Nitrogen at 4500 psi was not directly usable for the application due to the working pressure limitations of available lightweight solenoids. The nitrogen, therefore, had to be regulated to a lower pressure.

The regulator chosen for the thruster was also manufactured by PMI, the Pure Energy THOR Adjustable Regulator (Figure 14). It had the advantage of being adjustable down to around 100-120 psi (most paintball regulators are designed to regulate to higher pressures—usually 200 to 400 psi). This regulator also had the advantage of being easily mountable, and are pre-built with a bracket for installation. The regulator was easily modified for installation on SIMSAT. Once the nitrogen had been regulated down to 120 psi, it was suitable for pressurizing the thruster system.



Figure 14. PMI Pure Energy THOR Regulator.

Another component of the thruster system is the nitrogen valve solenoids, connected to the regulator via fluorinated ethylene propylene (FEP) tubing. The Mead Fluid Dynamics 2/2 Single Solenoid Valve was chosen for the thruster system (Figure 15). This valve offered several advantages. First, it was part of a modular design, which included a four station manifold and all electrical connections ported to a 15 pin sub-D connector. In other words, in order to fire a particular relay, it was only necessary to introduce a voltage potential difference of 12V to the appropriate pin pair. With each manifold capable of carrying 4 solenoids, it was then necessary to acquire two such manifold/solenoid banks. While only six solenoids would be required for 3-axis control, it was decided to order eight, to allow for future thrust scheme variations or as spares in case of malfunction. Each manifold/solenoid bank is rated for 120-psi pressure.

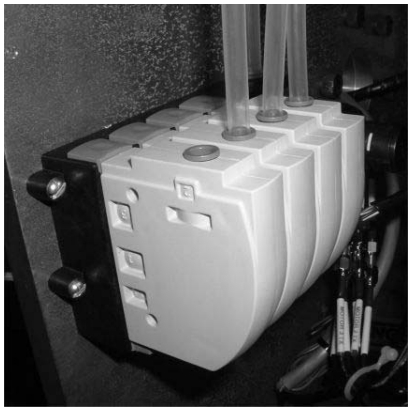


Figure 15. Nitrogen solenoid valves.

With the solenoids in place, the next step was to determine how to introduce the "open" signal to them from the computer. SIMSAT's onboard computer is connected to an I/O panel (Figure 16), which can provide analog output from the ds2103 board. Upon reviewing the board's specifications, it was found that it could provide a maximum of 10 V potential difference (which, in reality, proved to be closer to 5V) at low amperage.

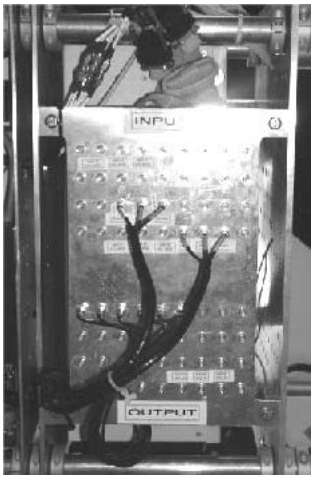


Figure 16. SIMSAT input/output panel.

To provide adequate power, it was necessary to introduce 5V normally open relays into the system, to pass the full 12V (and current) required of the solenoids, given

a 5V command from the board. The relay chosen for this application was the Aromat 5V PC Mount Relay, DSP Series Miniature. This relay would activate upon receiving a 5V signal, at a minimum current of 50 mA, and pass the 12V power supply on to the nitrogen solenoid valves. The 5V signal comes from the computer, based on the control logic for thruster firing. Six relays were mounted on a circuit board, one for each active solenoid nitrogen valve (Figure 17).

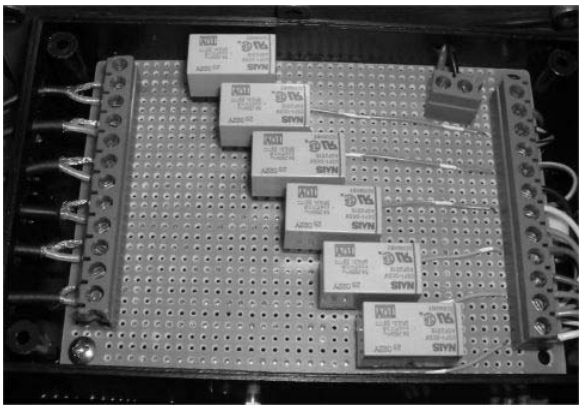


Figure 17. Relays mounted on circuit board.

Finally, the last step was to choose the nitrogen transport hosing. The FEP hose, mentioned above, is capable of transporting gas at a pressure rating up to 200 psi. The hose would be used to route nitrogen from the gas bottle and regulator to the solenoid system and then from the solenoid system to the thrust points.

Originally, the satellite was to be configured for three-axis control. For this design concept, there were going to be twelve thrust points—four ports per axis, two for positive and two for negative moments about each axis (Figure 18). For the b_2 and b_3 rotational axes, one of each thrusting pair was to be located on either longitudinal endplate, providing the maximum thrust moment arm. For the b_1 axis, the thrust points would have all been located on the end closest to the solenoids, to minimize the nitrogen

transport distance to the thrust points and the amount of air hose to be routed through the center sphere and arms. In Figure 18, each arrow indicates a thrust point and the

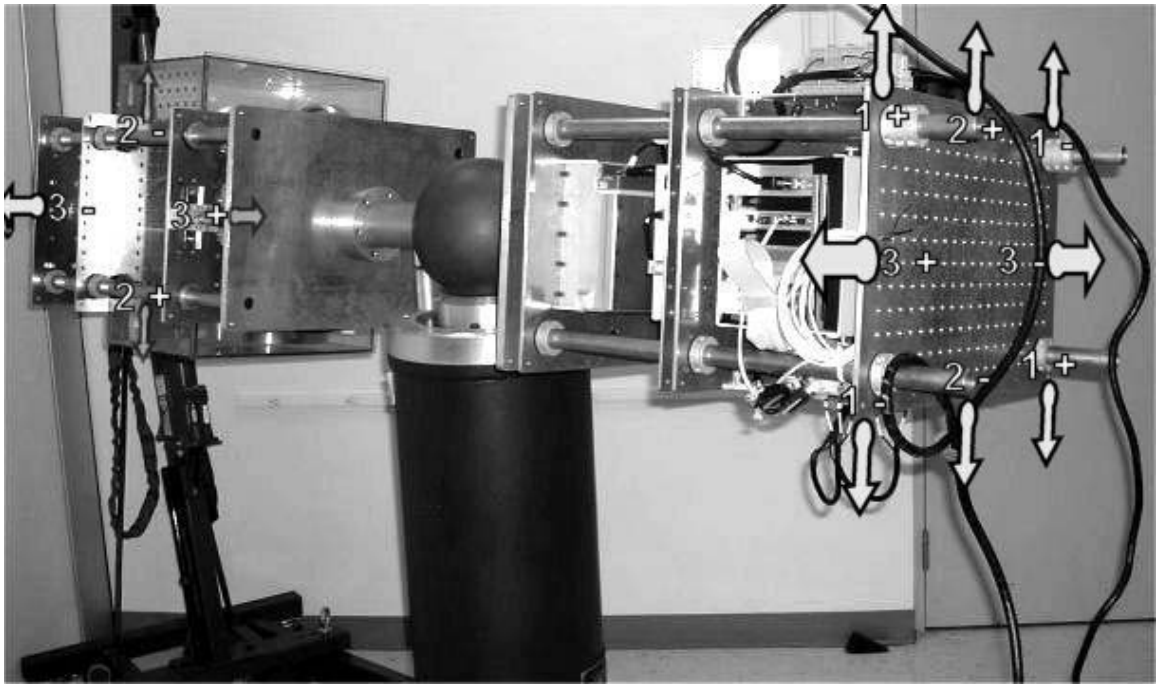


Figure 18. Original design thruster positions.

direction of thrust. Each point is labeled 1, 2 or 3, depending on the body frame axis it is controlling, and + or - depending on whether the moment created by that thrust-point is positive or negative with respect to defined body-fixed coordinate frame.

In the end, due to the problems already mentioned with pitch-axis gyro and roll-axis sagging, the experiment only sought to control the yaw axis. Also, instead of installing four exhaust ports per axis—two for positive yaw moment and two for negative yaw moment—two were installed, one positive and one negative (3+ and 3- in the left half of Figure 18). Only thrust one port was installed per yaw axis direction because it was determined that flow choking was occurring in the solenoids and consequently, more ports would be superfluous. The original control scheme was based on early tests using

compressed air at 100 psi to maneuver the satellite. By examining time-to-angle data, a rough expected thrust of 5 pounds was calculated. When the solenoids were sourced, however, it was clear this figure would not be realistic. The actual thrust obtained is discussed in Chapter 4, Open-Loop Thrust.

One final modification was made to the single axis control system. Given that six solenoids had been ordered to provide three-axis control and now only single-axis control was required, three solenoids each, rather than one, were assigned to the positive yaw direction and the negative yaw direction for a total of six solenoids, increasing the thrust.

With all of the elements of the thruster system selected, it was necessary to locate the system on SIMSAT. For several reasons, the rack on the side of the reaction wheels was chosen. First, this side had the most available room. The entire assembly could, if necessary, be located between the first and second plates on that side. Secondly, it created symmetry of functionality. All controls would be on one side, and other functions on the opposite. Third, the location of the solenoids close to the gyros was not desirable because the vibrations of the solenoids might affect the gyro output. The drawback to selecting that side was it would be necessary to route cable through the center sphere and arms. However, this was accomplished without much difficulty.

The only other note on the placement of thruster system subcomponents is the choice to place the relays on the computer/power bus side. This was done to facilitate timely construction and was seen as an arbitrary decision. A diagram of the final layout of the thruster system on the first radial plate on the control side is given in Figure 19.

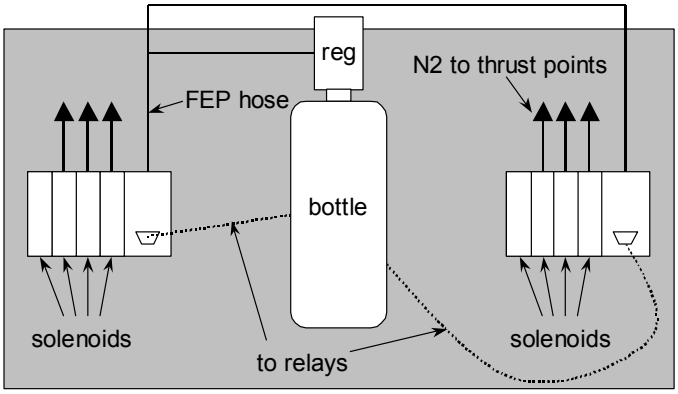


Figure 19. Thruster system layout on first radial plate.

Reaction Wheel Notes

The reaction wheels on SIMSAT possess the capability of providing SIMSAT with sufficient torque to provide relatively rapid slew rates. For the purposes of this thesis, the reaction wheels were relaxed. The wheel motor initialization routine includes setting the value of a parameter that controls the acceleration rate of the wheels given an wheel rate change command. The acceleration rate was reduced by one order of magnitude for the purposes of demonstrating a logical crossover point between thruster and reaction wheel control, given that reaction wheel control was desired mainly for the purpose of fine pointing.

Finally, the model of reaction wheel dynamics for the simulations run is accurate only for small demands. If large torques are demanded of the reaction wheel, the model begins to lose accuracy.

IV. Results

Overview

The results from testing will be presented in this chapter. First, there will be a brief discussion on how the experiments were conducted. Then results for the PD controller will be discussed, beginning with closed-loop reaction wheel control, then open and closed-loop thruster control followed by hybrid control. This will be followed by a discussion on model matching, including two actual vs. model comparison sets to demonstrate the fidelity of the model. Then the performance of SIMSAT will be discussed, including the trade-off between fuel expenditure and settling time and a the variation of gains to achieve minimum settling time over a range of desired final attitudes. Finally, the SDRE/LQR controller results will be presented.

Experiment Process

Experimentation on the SIMSAT was performed in a consistent manner. This process will be discussed in two parts: preparation and testing

Preparation

Preparing SIMSAT for tests began with turning on SIMSAT via three switches located on SIMSAT, one for each voltage level (12V, 24V and 36V). The 12V battery powers the thruster system—relays and solenoids. The 36V battery powers the reaction wheels only. The 24V battery powers everything else.

Next, it was necessary to link SIMSAT's Autobox computer with the "ground station" computer via the RadioLAN® wireless network. Once SIMSAT Autobox computer was on and the ground station computer booted up, the Control Desk software could be started up, and it would automatically look for the Autobox via the LAN.

When this was accomplished, the relevant software had to be loaded into the dSpace work environment for running the test and data capture. This included any dSpace layout files for display and capture of data (pd_dual2.lay, for example), any Simulink model files one wished to modify (PD_dual2.mdl), and any other supporting software for setting up the model (simlaunch6.m) or formatting the data for export (data_get.m). Many of the Simulink .mdl files are available in Appendix A. The listings for the above mentioned m-files are available in Appendix B.

Testing

After loading required software, SIMSAT was allowed to float free by removing the pitch/yaw restraints (Figure 20) after detaching any ground station test power cables and battery recharge cables that were attached.



Figure 20. SIMSAT pitch/yaw restraints.

With the satellite mobile, it was possible to begin testing. Each time a test was run, SIMSAT would be aligned such that all three axes were oriented began in approximately the same position (b_1 axis pointing towards the back of the lab, away from the door, and b_3 axis pointing up in the local vertical). It was also necessary to ensure SIMSAT remained essentially stationary. Finally, the pressure reading on the nitrogen

regulator had to be checked, to ensure it was reading approximately 120 psi. If it wasn't, the regulator was adjusted via a hex key adjustment on the top of the regulator until it did.

Once this was achieved, the Simulink model (PD_dual2.mdl) would be compiled and loaded into the Autobox computer. As it compiled, a process that usually took around 20 seconds, the layout file would be prepped for activation soon after the compiler indicated that the model file was "loading." The position, thruster activity, wheel activity and system energy would be displayed via layout (Figure 21), and after the run was complete, the captured data could be saved.

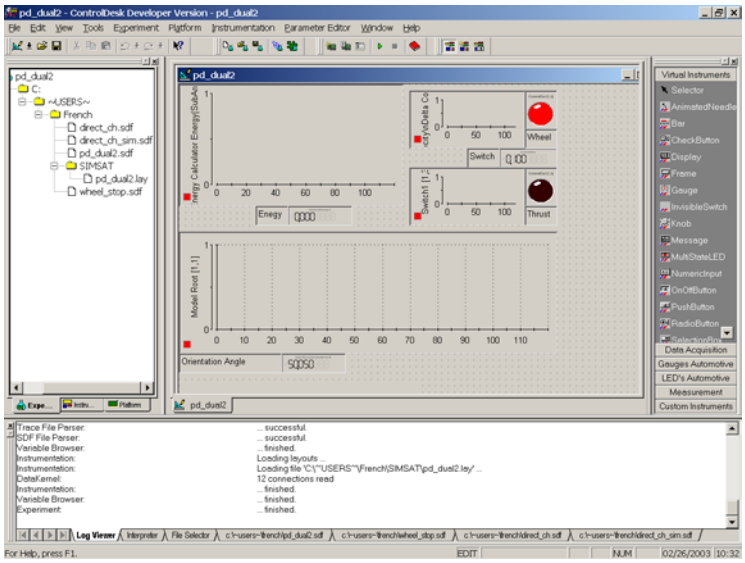


Figure 21. ControlDesk layout

Having introduced the test method, it is now necessary to discuss the experiments performed on SIMSAT, beginning with the earliest—closed-loop wheel control.

Closed-Loop Reaction Wheel Control

Before the thrusters were installed, a simple algorithm was run to begin to match the performance of the model to test results using only the yaw-axis wheel for control.

The model was tasked with following a square wave input. Then the same task with the same gains was tested experimentally (Figure 22).

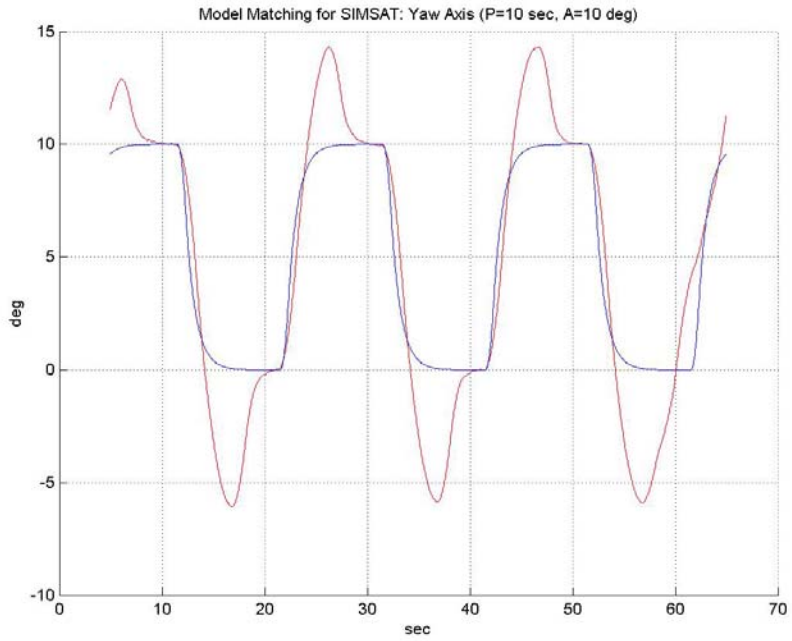


Figure 22. Initial Results for reaction wheel matching.

In this chart, the inner line—the one that closely approximates the square wave—represents the model. The actual results overshoot, and the initial attempt at modeling the system needed tuning. Using the scaling factors, $K_{motor,in}$ and K_{motor}, the hysteresis curve (Figure 10) describing the wheel response to control demands was adjusted to fit it the experimental results. The final values of the two gains for the final model were found to be $K_{motor,in} = 1$ and $K_{motor} = 0.1$. With these values, a similar test was run to test wheel control tracking (Figure 23).

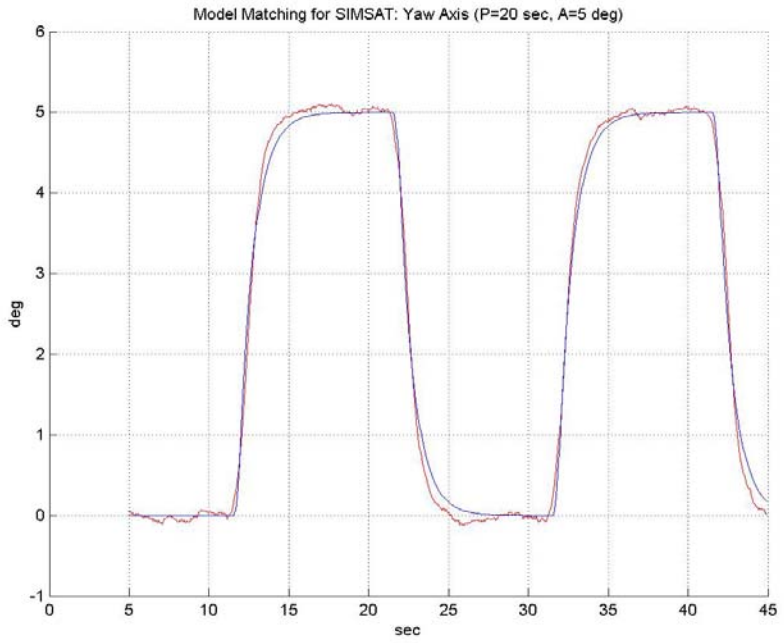


Figure 23. Matched wheel response.

For these conditions, the model matches well to the experimental data. The range of applicability of the motor model was found to be limited late in testing, however, especially when more than fine pointing was demanded of the motor. See the Hybrid Control section later in this chapter for more details.

Open-Loop Thrust

After the thrusters were installed, one of the first results of its control was how SIMSAT responded to an open-loop thrust command (Figure 24). This test was accomplished before the system was modified to include three solenoids per yaw axis direction. At this stage, there was only one solenoid per yaw axis direction. The yaw axis thruster was activated for 10 seconds. The change in yaw rate was noted by measuring the yaw rate of change before the thrusters were fired—the slope of the line while the thrust indicator is still zero—and then measuring the slope again after the thrusters had stopped firing. The thrust was assumed to be constant throughout the pulse.

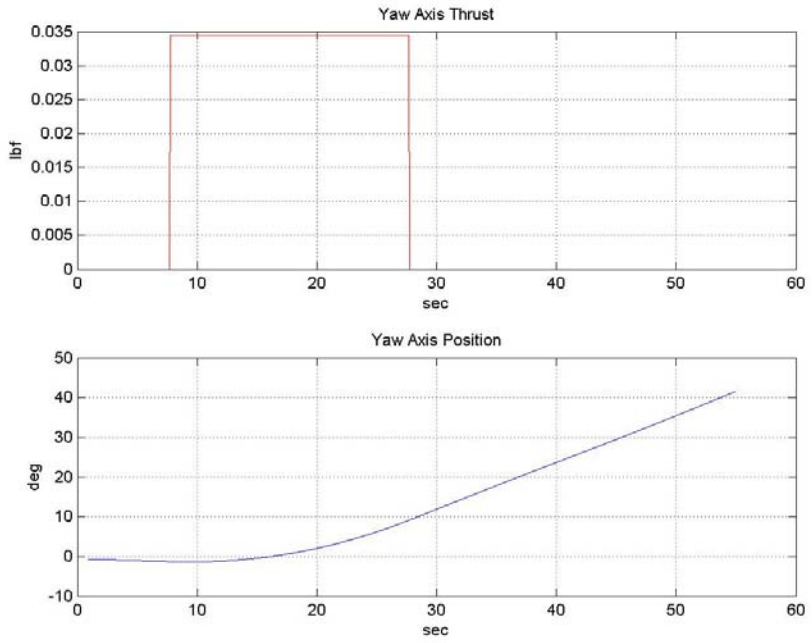


Figure 24. Open-loop thrust command response.

With a few simple calculations, the average thrust of the thrusters could be determined. First, a relationship between moment and angular rate had to be established.

$$M = I \cdot \frac{d\omega}{dt} \tag{37}$$

$$\int_{t_0}^{t_f} M\,dt = \int_{t_0}^{t_f} I \cdot \frac{d\omega}{dt}\,dt = \int_{\omega_0}^{\omega_f} I\,d\omega \tag{38}$$

$$M \cdot (t_f - t_0) = I(\omega_f - \omega_0) \tag{39}$$

Therefore, given that everything in Equation 39 except M is known, and given that

$$M = T \cdot d \tag{40}$$

the equation for thrust reduces to

$$T = \frac{I \cdot (\omega_f - \omega_0)}{(t_f - t_0) \cdot d} \qquad (41)$$

In this case, the thrust derived turned out to be approximately 0.035 lbs per solenoid. This value is given in lbs per solenoid, because it was found that opening three solenoids did, in fact, triple the thrust produced, as later model matching would verify. This value of 0.035 lbs turned out to be near the bottom of the range of thrust values obtained throughout the experiment. The most common value was approximately 0.045 lbs per solenoid, but values ranged as high as 0.070 lbs per solenoid on occasion. These higher values for thrust occurred more as the tank's pressure dropped. The reason for this pressure drop is unknown and remains as a topic for future study.

Closed-Loop Thrust Control

With the open-loop data analyzed, closed-loop control using the thrusters was sought (Figure 25). Closed-loop control was attained using a single solenoid for the positive yaw thrust input and a single solenoid for the negative yaw thrust input. The reason for switching to three solenoids per yaw axis direction is clear when examining the above chart. For a maneuver of only 5 degrees, the settling time was around 30 seconds.

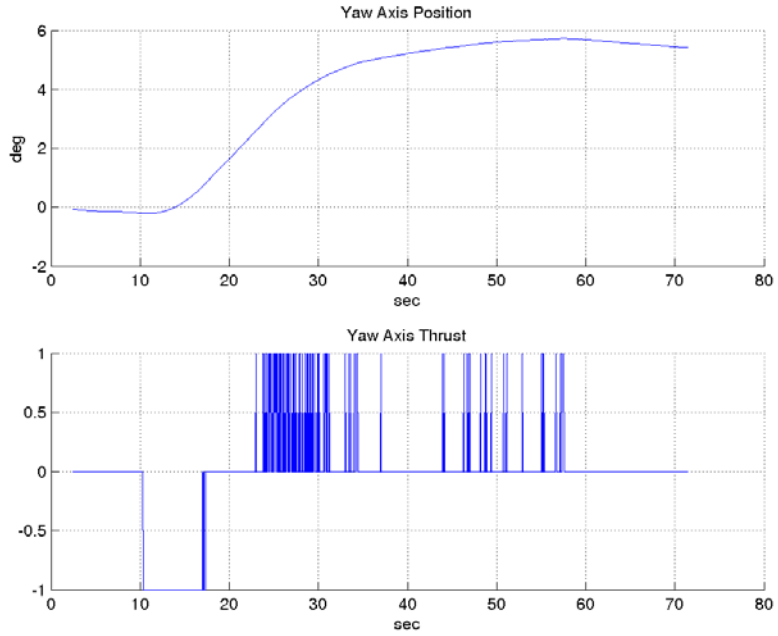


Figure 25. Closed-loop thrust control.

With closed-loop wheel control and closed-loop thruster control successfully achieved, hybrid control could be sought. SIMSAT was modified with the energy switch to form the hybrid controller.

Hybrid Control

The reaction wheels are intended to increase pointing accuracy, but the question of how a higher energy exchange threshold for switching between reaction wheel and thruster controls affects settling time and fuel expenditure is still valid. In order to answer this, it was necessary to examine how performance varied with the energy exchange threshold.

Control Switching

The control switch energy level (CSEL) is the "energy," as defined in Chapter 3, Control Switching, at which attitude control is switched between thrust control and reaction wheel control. Recalling equations 34 through 36, it can now be defined that for

E > CSEL, the thrusters are active, and for E < CSEL, the reaction wheels are active. Figure 26 gives an example of control switching. Notice that the energy level crosses the threshold several times over the testing period, and control switches between wheels and thrusters at those times.

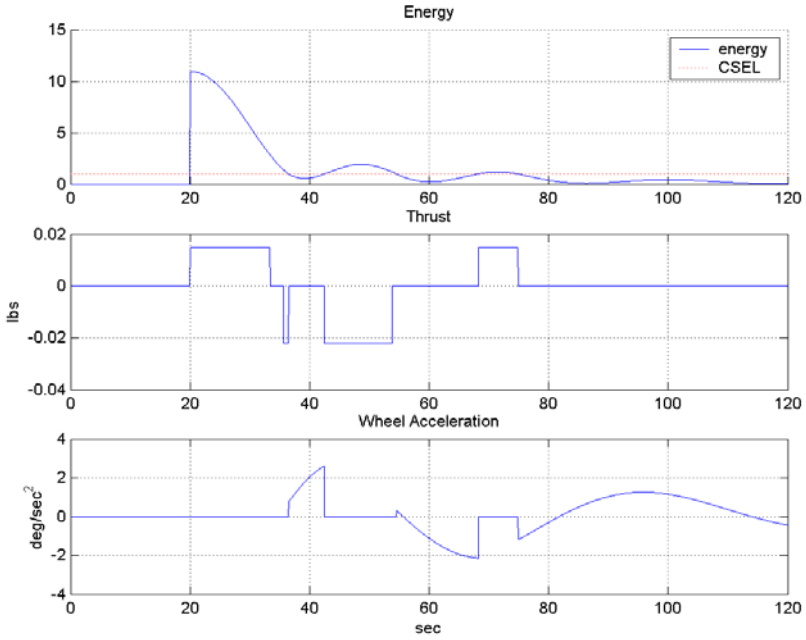


Figure 26. Control switching

The results for a wide range of CSEL settings are available in Figure 27.

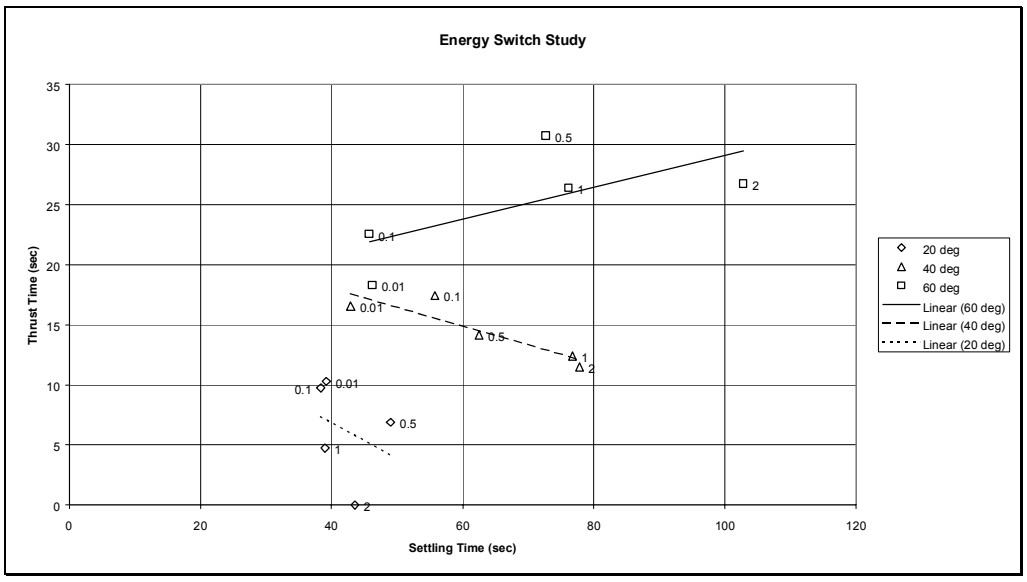


Figure 27. Energy Switch Study.

The figure gives performance of SIMSAT for slew ranges of 20, 40 and 60 degrees, with energy handoff points of 0.01, 0.1, 0.5, 1 and 2. The first tendency to note is that settling time decreases with decreasing energy switch values, as expected. This is due to the thrusters having a higher capacity for creating torque than the reaction wheels. The second tendency to note is that the thrust durations increase for the 20 and 40 degree slew cases for a lower energy exchange value. This also seems intuitive, as the thrusters affect more of the control and thus use more fuel. For the 60-degree case, however, there exists a decreasing trend with decreasing energy handoff value.

To explain this, compare Figure 26 to the following chart (Figure 28). Figure 26 shows the model handing off control at high energy, while Figure 28 shows a low-energy transfer.

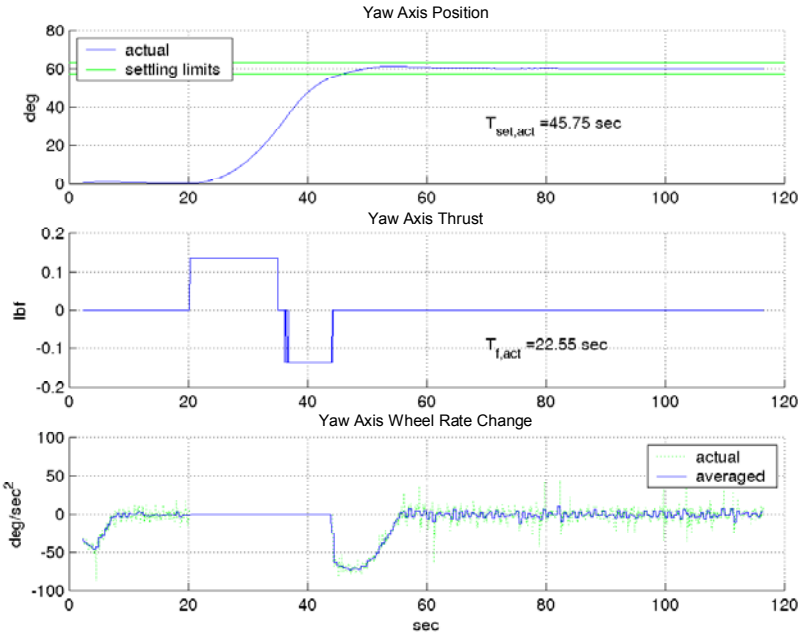


Figure 28. Normal handoff from thrusters to wheels

This figure demonstrates a normal handoff of control for a high-performance slewing maneuver. Compare the bottom two charts on the figure—the thrust chart and the wheel acceleration chart. At first, the wheels are controlling attitude, because SIMSAT is at a low energy state (attitude demand does not change until 20 sec). At 20 seconds, the thrusters activate and remain active until they pass control back to the wheels, after which, the wheels maintain control of the attitude, as desired.

In contrast, Figure 26 showed a high-energy hand-off from thrusters to wheels. The transfer of momentum from SIMSAT to the wheels does not occur quickly enough, and the thrusters reactivate to assist in recovery. This results in a higher amount of work done by the thrusters than if they would have been used to bring SIMSAT to a lower energy state before transferring control to the wheels. Overall, CSEL settings between 0.05 and 0.1 worked best and were used for all data collected.

Fine Pointing

The contribution sought from the reaction wheels is stability at a given pointing angle. As established in Chapter 2, the reaction wheels provide much greater pointing accuracy. Over the course of data collection, the reaction wheels provided a pointing accuracy of less than 1-degree error, once SIMSAT has settled to a low energy state—for most cases, after around 90 seconds (70 seconds after slew initiated).

When performance data is discussed in upcoming sections, settling time is defined as the time at which the angular position delta has closed to within 5 percent of its original value. After this goal had been reached, it was determined that SIMSAT would still oscillate at a low level about its target. The largest deviation from pointing target is shown below in Figure 29. In this figure, the angular position was examined after SIMSAT had settled well within its 5 percent boundaries. Oscillations varied by almost an order of magnitude over the test cases examined.

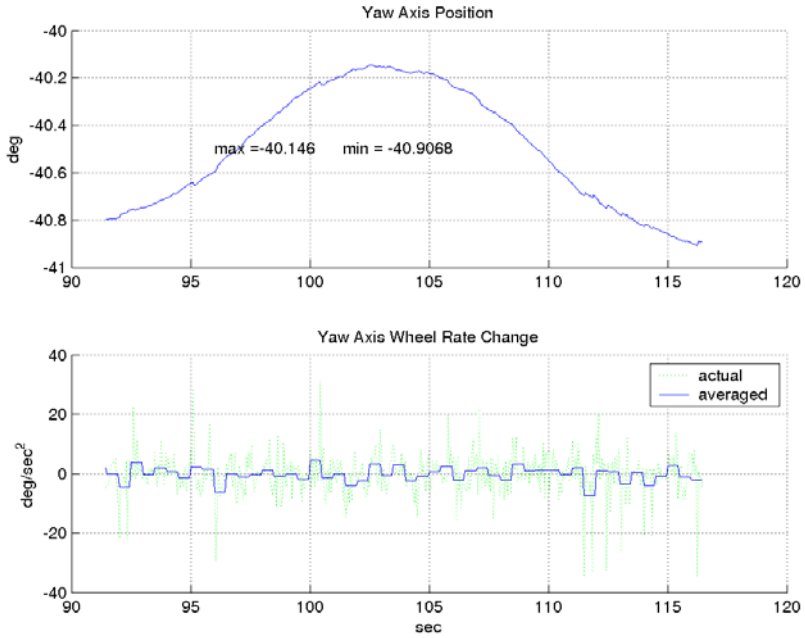


Figure 29. Worst case pointing error.

One possible explanation of the oscillation is that it may be a result of the controller tracking the slow drift of the yaw gyro. Additionally, note the level of noise in the wheel inputs. This noise corresponds directly with the noise in the gyros. Developing strategies for reducing pointing error remains as a topic for future study. A good first step would be too implement a low band pass filter.

Model Matching

The model for SIMSAT was refined throughout the testing process. When discrepancies were found between experimental results obtained and the model, a reason for the discrepancy was sought, and the model adjusted.

Figure 30 shows an early model and actual data for a 45-degree maneuver. Before discussing these results, a brief explanation of the notations is necessary. The top chart shows displacement with respect to time. The notation $t_{set,mod}$ stands for "computer model settling time" and $t_{set,act}$ stands for "actual settling time." The bands shown just above and below the target attitude span angles within 5 percent of the target attitude.

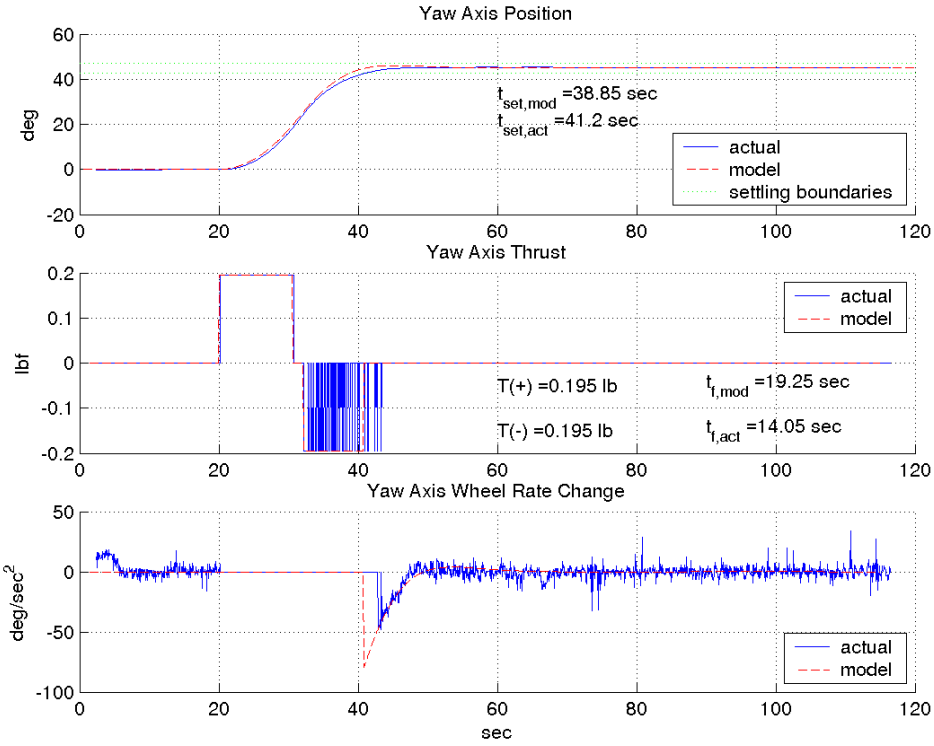


Figure 30. Thrust matching assuming symmetric thrust.

One final note is necessary. For the charts given from this point on, when a slew maneuver is performed, it is begun at the 20-second mark. This is to allow the wheel control to first stabilize SIMSAT and fix it to the 0 degree orientation, according to its gyro. Values of settling time include this 20-second stabilizing period. The middle chart gives thrust with respect to time. Positive yaw thrust and negative yaw thrust are indicated as T(+) and T(-), respectively. Note that these values are the combined total of thrust, not thrust per solenoid. The amount of time thrust is being expelled is also given—$t_{f,mod}$ for the computer model and $t_{f,act}$ for actual results. The third chart shows reaction wheel acceleration demand with respect to time.

With this basic understanding of the figure, discussion of what it shows may commence. In this figure, the model assumes the thrusters are symmetric—each

providing equal thrust. It can be seen in the figure that this was not the case. While accelerating in the positive direction, the model tracks the actual results. However, when the model's deceleration thruster is fired, it cannot match the performance demonstrated experimentally, even though it is firing continuously while the actual thruster pulsed. It was determined that the thruster system has a bias toward the negative thruster—thrust in the negative direction exceeds that in the positive by approximately 25 percent. This, too, was programmed into the model.

The reason for this bias is not certain. One possibility is the difference in length of the expellant path to the thrust points (Figure 31).

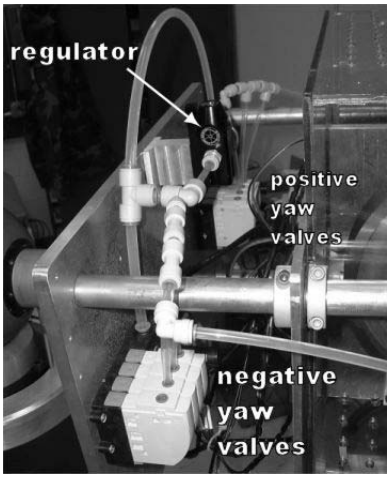


Figure 31. Potential cause for thruster bias.

After leaving the nitrogen bottle and passing through the regulator, the nitrogen must travel through approximately 2 feet more tubing to reach the positive yaw thrust valves than to reach the negative yaw thrust valves. After the nitrogen reaches the solenoid air valves, the travel distance to the thrust point is approximately equal. However, this explanation seems unlikely as the cause, given that the cross-sectional area of the tubing that connects the regulator to the solenoid valves is nearly 4.5 times as large as the

combined orifice areas of the thrust solenoids. The actual cause of this discrepancy remains as a topic for future investigation.

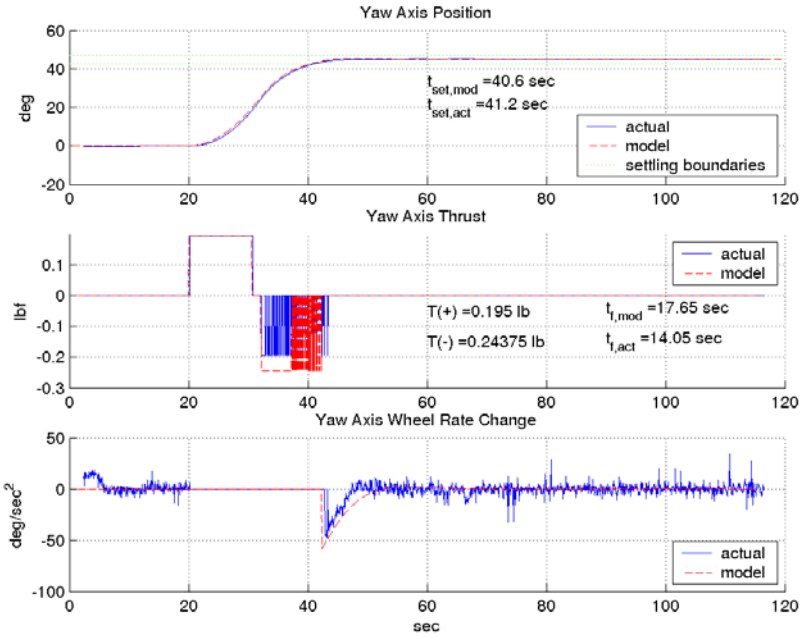


Figure 32. Thrust matching including 25 percent negative bias

Figure 32 includes the thrust bias. With this accounted for, the performance of the model approaches perfect emulation of the actual system for this case. Note that in the middle chart the actual thrust magnitude does not match the model thrust magnitude for negative thrusts. This is due to thrust bias. The model accounts for thrust bias, because the simulation is aware it is feeding a higher thrust to the negative torques. The actual thrust data is a vector of the discrete values –1, 0 or +1, which command the thrusters to thrust negative, remain off, or thrust positive, respectively. This vector is scaled by the positive side torque found in the model matching process for the purpose of comparison.

One final note on Figure 31 is the noise level on the actual wheel acceleration curve. This is due to the noise level of the gyro. For the relatively small commands passed to the reaction wheel motors, this noise, as seen, is significant.

The next two subsections detail the bulk of the model matching data collected. The first series was run with constant gains set for the thrusters and wheels on both experimental results and modeled results. The second series was run for constant angles and varied gains. The purpose of this data is to increase confidence in SIMSAT model.

Constant Gains – Varied Angle Matching

Having accounted for the thrust bias in the model, it was necessary to test SIMSAT over a wide variety of angular displacements and gains (Figure 33). The first series of model matching experiments maintained constant gains for thruster and reaction wheel control, varying only the displacement angle sought. The angles would include +/- 20 deg, +/- 40 deg, +/- 60 deg and +80 deg.

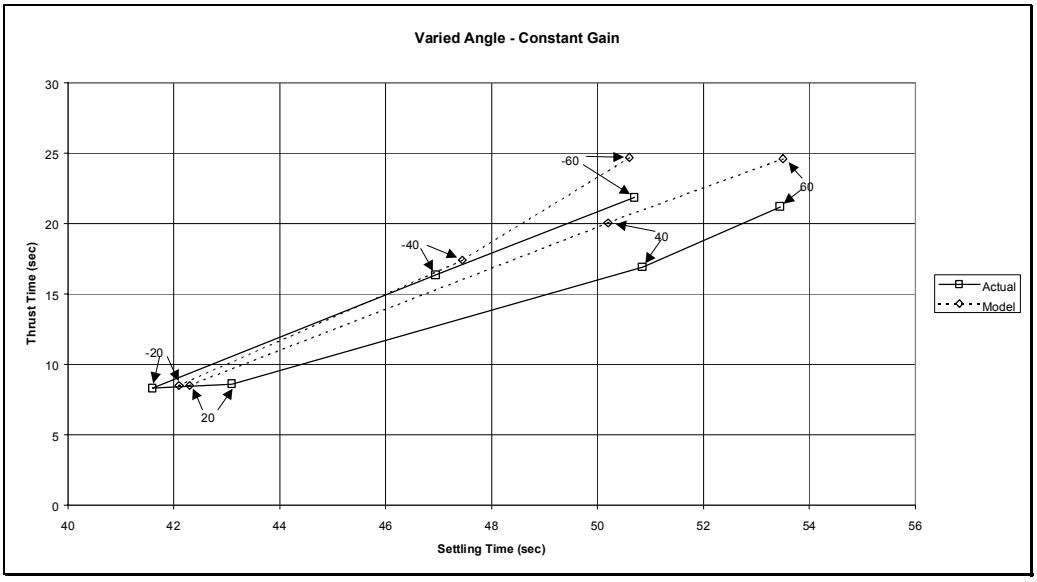


Figure 33. Constant Gain vs. Varied Angles

As can be seen, the model is similar in performance to the experimental results. Once the experimental data was obtained, each was matched with the model using only one "tuning knob," that being the thrust on the positive side, which generally remained around 0.045 lbs per solenoid, for a total of 0.135 lbs. The negative side thrust would then be extrapolated via the bias constant of 1.25 discussed above. Notice the negative angular goals are reached quicker due to the thrust bias.

The requirement for the thrust to be adjusted each run stems from the lack of precision of the regulator used. The regulator was unable to produce consistent pressures over time.

The first of these runs was conducted for a displacement of 20 degrees (Figure 34).

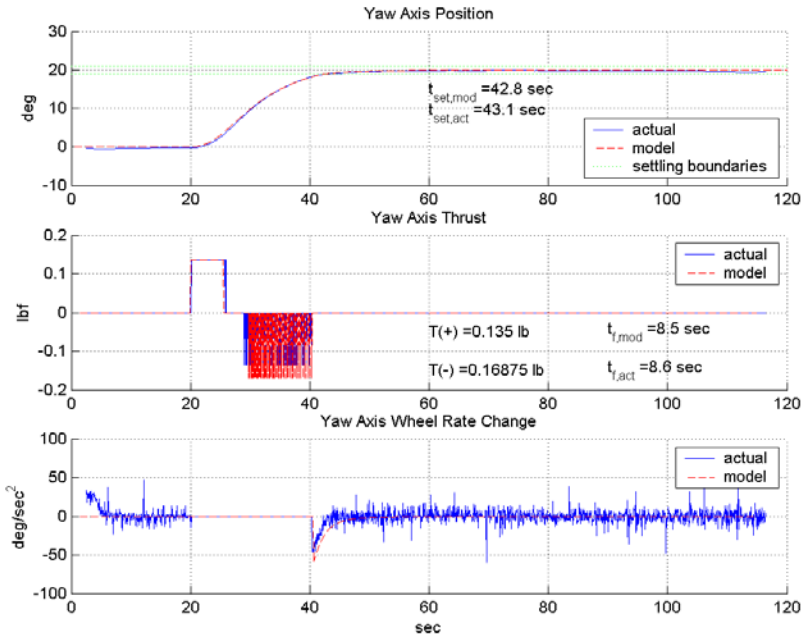


Figure 34. Displacement of 20 degrees.

This first case shows good agreement between the model and the experimental case. The model matched the experimental results for a positive side thrust (T_{s+}) of 0.045 lbs per solenoid.

The next case was for a rotation of negative 20 degrees (Figure 35).

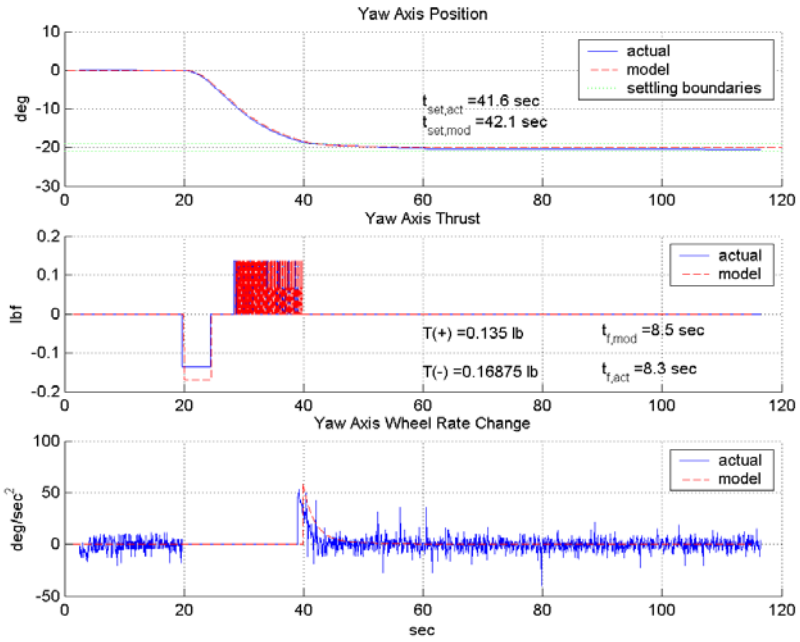


Figure 35. Negative 20-degree rotation.

For this case, T_{s+} was again set to 0.045 lbs.

The next case was for a rotation of 40 degrees (Figure 36).

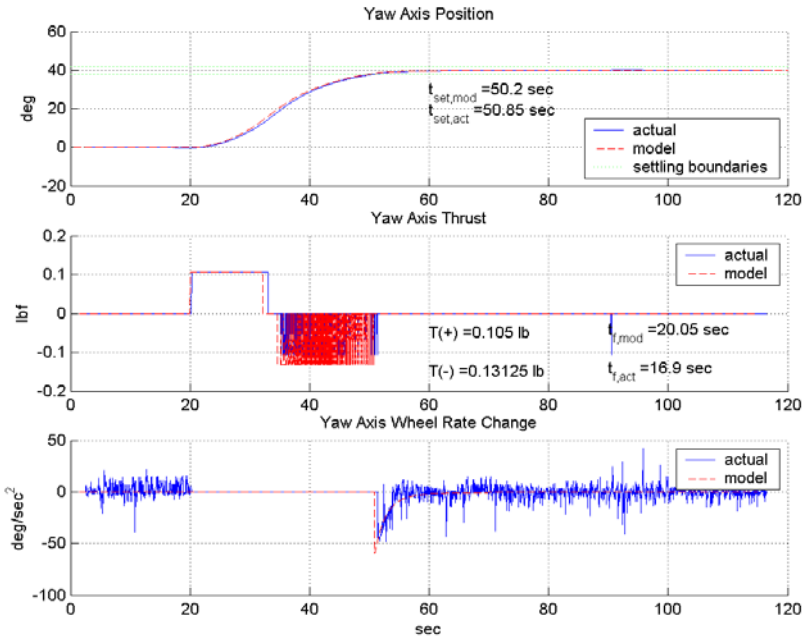


Figure 36. Positive 40-degree rotation.

One note on this case is the thruster firing at approximately the 90-second point. Occasionally, the gyro noise would peak at some high values, causing the energy setting to surpass its threshold and fire. This could be remedied by either setting the energy threshold higher, using a low-pass filter on the gyro (see document by Fulton) or, preferably, replacing the gyro. Otherwise, the model matches the data well. Note, however, that T_{s+} for this case was 0.035 lbs per solenoid.

The next case was for a rotation of negative 40 degrees (Figure 37).

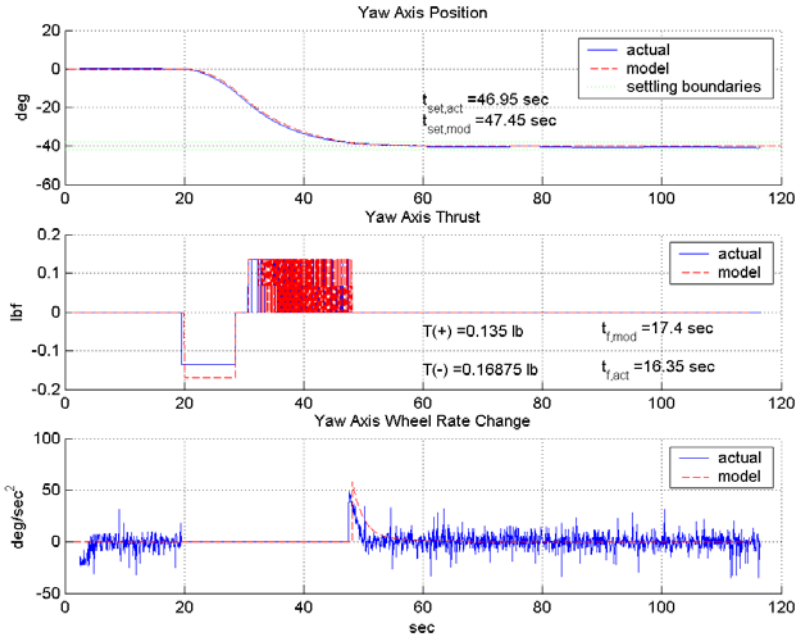


Figure 37. Negative 40-degree rotation.

Thrust for this case is back to the average; T_{s+} equals 0.045 lbs per solenoid.

The next case is for a positive 60-degree rotation (Figure 38).

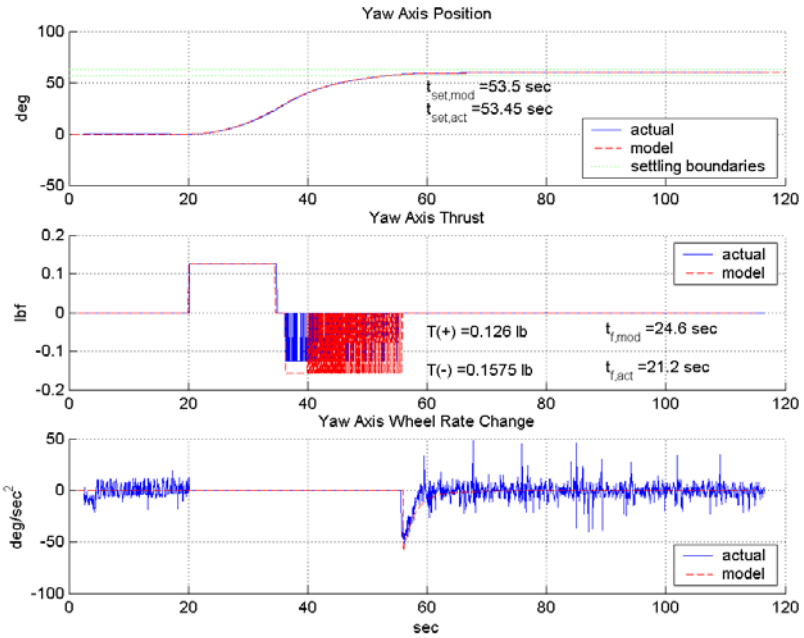


Figure 38. Positive 60-degree rotation.

For this case, T_{s+} fell slightly to 0.042 lbs per solenoid.

The next case sought an angular displacement of negative 60 degrees (Figure 39).

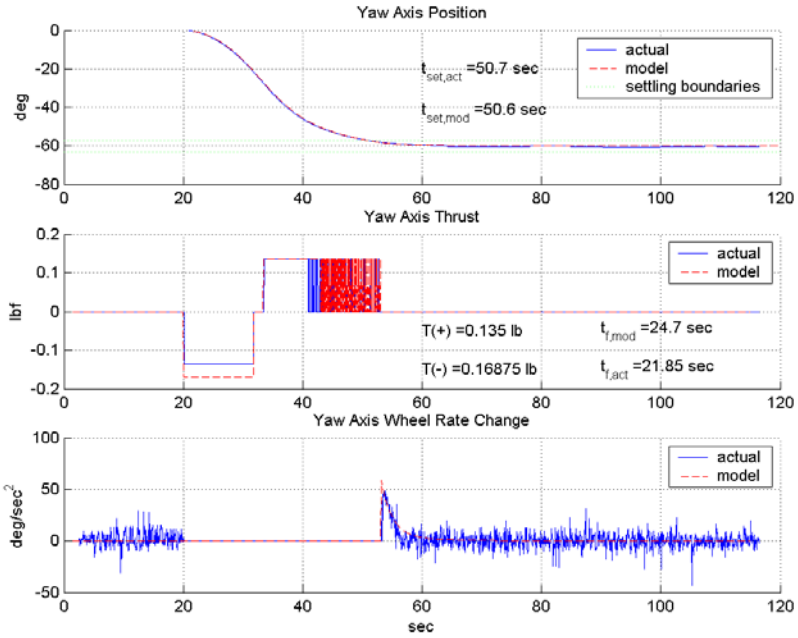


Figure 39. Negative 60-degree rotation.

Again, T_{s+} is back up to 0.045 lbs per solenoid with good model matching.

The last case is for a positive 80-degree rotation (Figure 40).

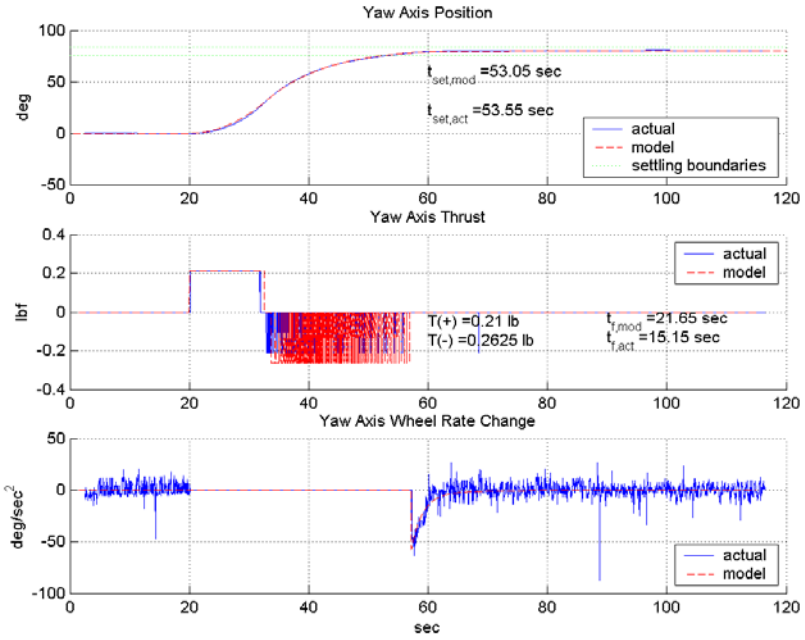


Figure 40. Positive 80-degree rotation.

This was the last usable test run. As the bottle pressure fell below 1000 psi as it did for this run, notice that the thrust increased to 0.070 lbs per solenoid. Why this increase occurs is unclear and remains a topic for future investigation. Note also that the model matches well to the experimental data for this, the largest case in the run. For more detail on the constants, gains and results for this series of tests, refer to the table in Appendix C.

Constant Angle – Varied Gains Matching

For this second series of tests, the angular goal was set to 60 degrees, and the gains were varied. Specifically, the rate gain associated with the thrusters ($K_{T,rate}$) gain was adjusted, and the rate gain associated with wheel control ($K_{w,rate}$) was adjusted by the same proportion automatically by a scaling constant.

$$K_{w,rate} = K_{T,rate} \cdot k_{scale} \qquad (42)$$

This series of test was run for $K_{T,rate}$ gains of 10, 15, 20, 25 and 30 (Figure 41).

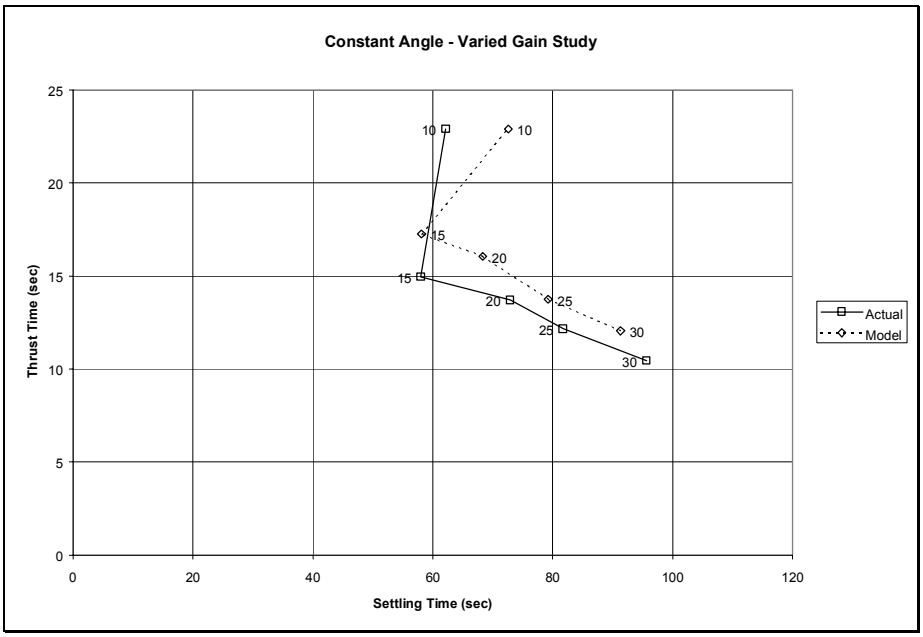


Figure 41. Constant Angle – Varied Gain

For this series of data, the model matched the experimental data well, until the low end of the gain spectrum. This occurred due to the limits of the reaction wheel model. The reaction wheel model is based around the concept that it will only be required to provide fine pointing. Large demands of the wheel are therefore not well matched by the model.

The charts below present the model matching comparison runs. The first of these tests occurs for a thruster rate gain of 10 (Figure 42).

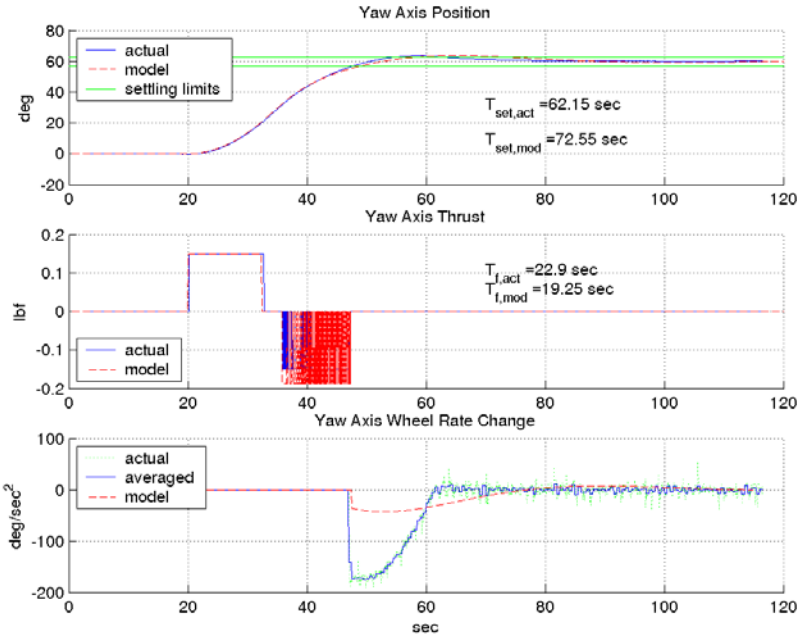


Figure 42. Thruster Rate Gain of 10.

As noted, the wheel acceleration results are not well matched by the model.

The next case is for a thruster rate gain of 15 (Figure 43).

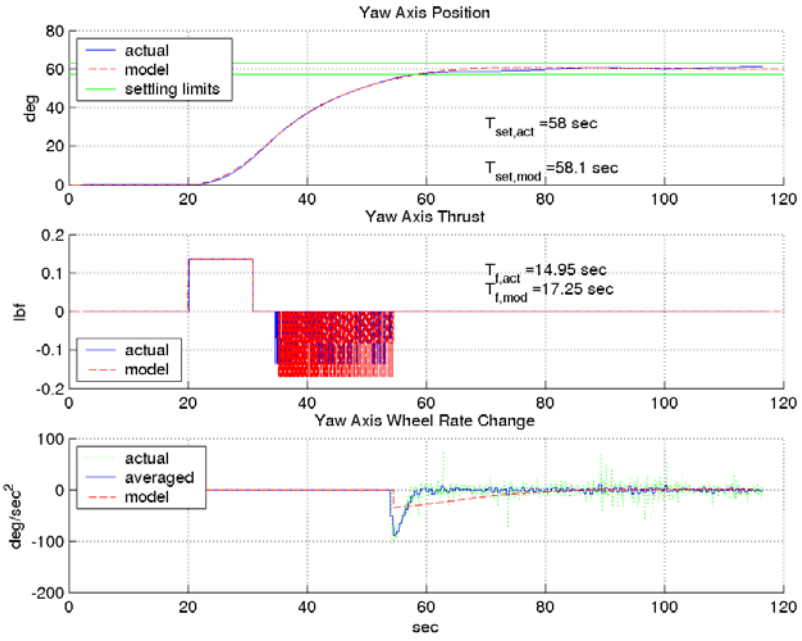


Figure 43. Thruster Rate Gain of 15.

The next case is for a rate gain of 20 (Figure 44).

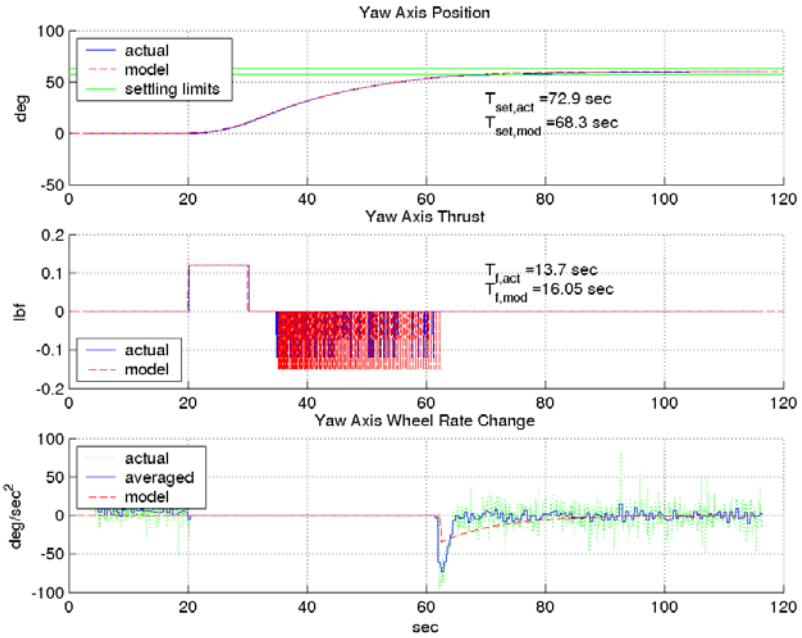


Figure 44. Thruster Rate Gain of 20

The next case is for a thruster rate gain of 25 (Figure 45).

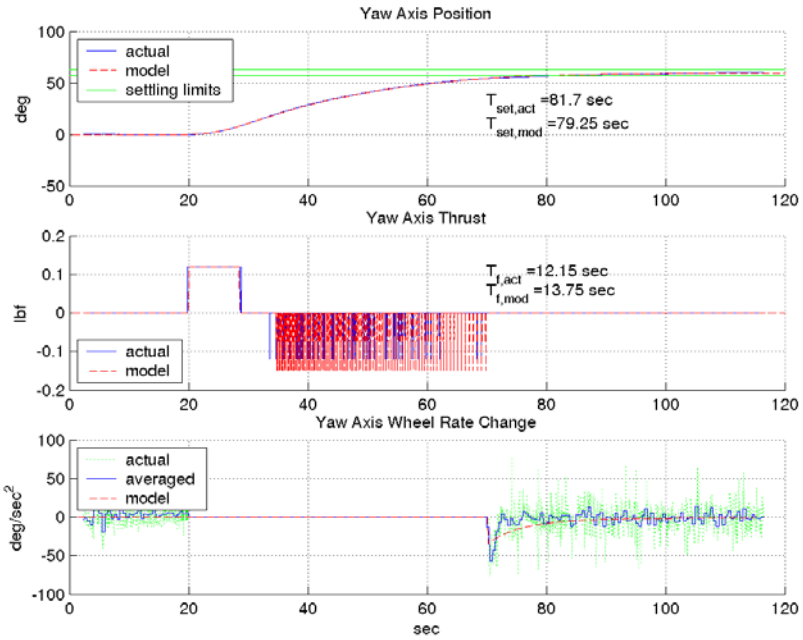


Figure 45. Thruster Rate Gain of 25.

The final case in this run is for a thruster rate gain of 30 (Figure 46).

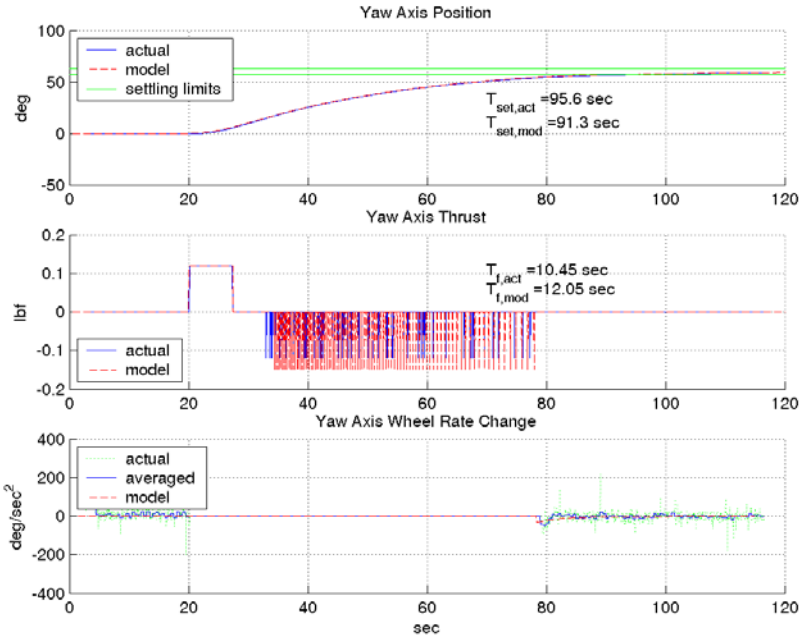


Figure 46. Thruster Rate Gain of 30.

Maximum Performance Study

With the confidence gained in the model, it could now be used to predict performance over a wide scale of angles and gains. The settings would be consistent throughout the series of runs, with the exception of the thruster rate gain varying each run. A sample list of settings is provided below in Table 3.

th_target	Target Orientation (deg)	45
Control Variables		**Values**
ep_K_T	thruster dead band	0.05
K_r_T	thruster rate gain	5
K_d_T	thruster delta gain	1
K_o_T	thruster overall gain	1
K_r_W	wheel rate gain	5000
K_d_W	wheel delta gain	1000
K_o_W	wheel overall gain	-1.1
eswitch	control switch energy	0.015
System Match Settings		
K_motor	motor out gain	0.1
K_motor_in	motor in gain	1
T_bias	thrust bias (negative stronger)	1.25
Performance Measures		
T_av=.035		
T_s (sec)	Settling Time [5% criteria] (sec)	63.6
T_f (sec)	Thrust Time (sec)	42.65
file		
T_av=.045		
T_s (sec)	Settling Time [5% criteria] (sec)	55.6
T_f (sec)	Thrust Time (sec)	33.7
file		
T_av=.055		
T_s (sec)	Settling Time (5%)	49.15
T_f (sec)	Thrust Time	27.4

Table 3. Sample data for thrust vs. fuel expenditure tests.

The rate gain was varied between 5 and 30, and separate curves were generated for three values of thrust: 0.035 lbs per solenoid, 0.045 lbs per solenoid and 0.055 lbs per solenoid (Figure 47).

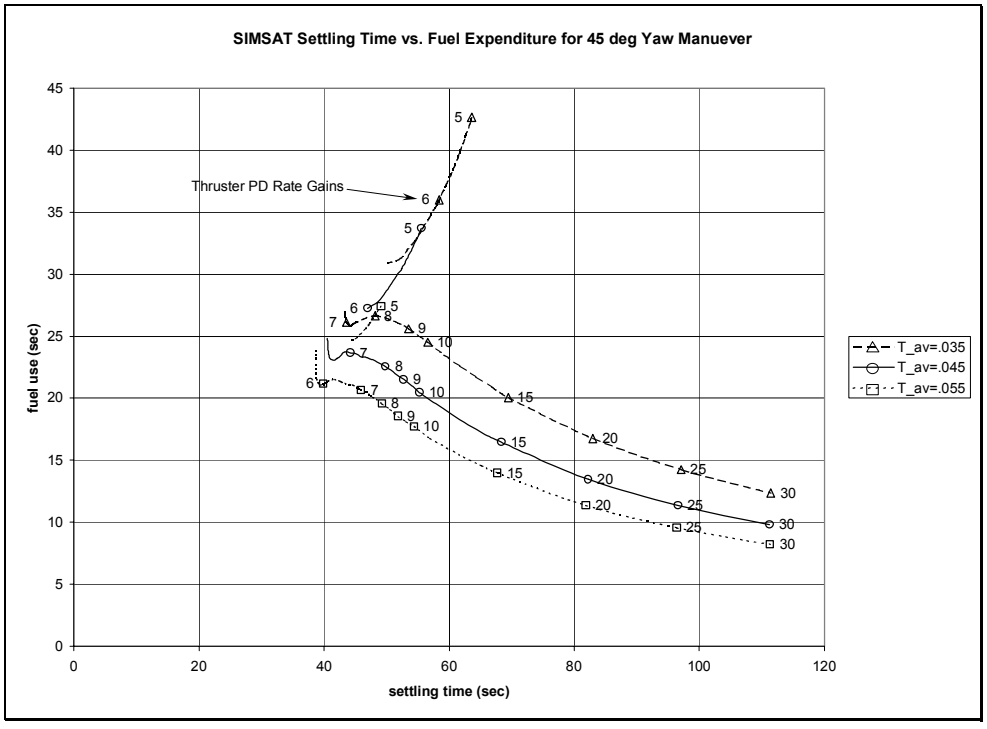


Figure 47. Settling Time vs. Fuel Expenditure

The maximum performance points can be seen clearly for this chart, at the gain for which the upper curve ends and the lower curve begins. This jump occurred when the gain caused the system to respond in such a way that the overshoot did not exceed the 5 percent boundary. For the 0.045 lb per solenoid curve, for example, the settling time is 40.55 sec for a thruster rate gain of 6.04.

Next, it was desirable to determine how fast the satellite could maneuver to a given angle. Consequently, the model was tested at angles of 10 to 90 degrees, in 10 degree increments, to determine the minimum settling time by adjusting the thruster rate gain (Figure 48). At each point, the PD rate gain used to achieve this minimum settling time is shown.

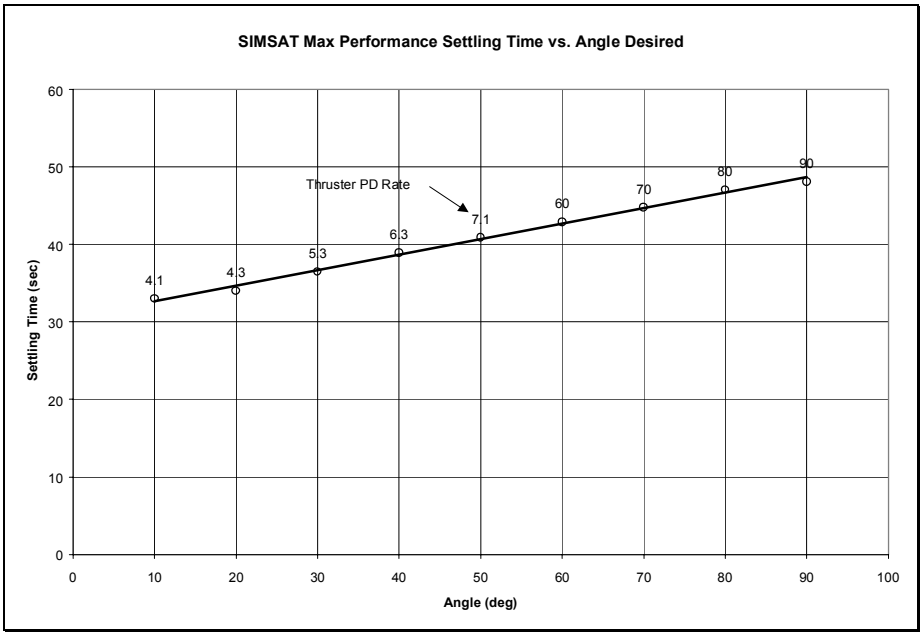


Figure 48. Minimum settling time vs. rotation angle.

The relationship over the range of angles tested was determined to be approximately linear. The equation describing the relationship between settling time and angular position is given in Equation 43.

$$t_s = 0.1996 \cdot \theta + 30.671 \qquad (43)$$

In summary, the performance of the PD controller is a function of many settings. In order to perform an arbitrary slew maneuver, the CSEL was set between 0.05 and 0.1. If a maximum performance (minimum settling time) slew was desired, it was necessary to interpolate the rate gain setting from Figure 48, depending on the angle desired (the position gain in all cases was 1). Any rate gain higher than the interpolated value would decrease fuel expenditure at the cost of performance.

SDRE/LQR Control

The SDRE controller was not applied to the physical SIMSAT, but the final computer model of SIMSAT was fitted with the SDRE controller to highlight a "what if" scenario. Again, SIMSAT model used is not exactly like SIMSAT. The model assumes SIMSAT is rigid and provides accurate full-state feedback. As discussed earlier, the long axis sagging and the pitch axis gyro preclude this from being the case. Even though the model allows for accurate predictions for the single-axis yaw case, the predictions it would yield for roll and pitch, or more accurately b_1 and b_2 axis, control would be off the mark.

With the SDRE controller applied to the idealized SIMSAT, it was found that for any given initial condition, given an appropriate choice of Q and R matrices, SIMSAT would theoretically always perform the maneuver successfully.

The case shown in Figure 49 is an example of the control the SDRE controller provided for the computer model.

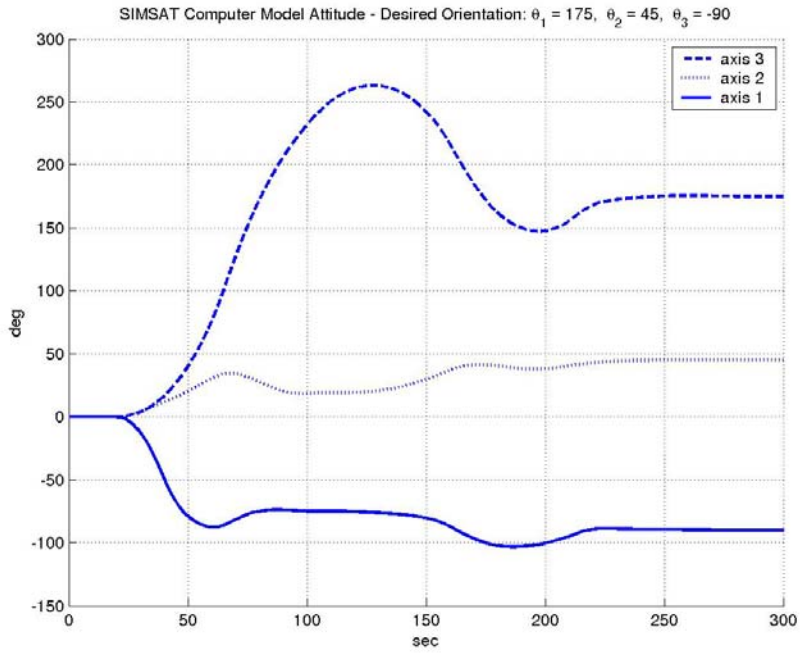


Figure 49. Results of SDRE control applied to SIMSAT computer model.

In this case, θ1, θ2 and θ3 have been assigned target values of 175, 45 and –90 degrees respectively. The weighting matrices Q and R are given below in Equations 44 and 45 for each of the sub-controllers. The weighting matrices are subscripted "T" for the thruster sub-controller, and "W" for the wheel sub-controller.

$$Q_T = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{bmatrix} \quad (44)$$

$$R_T = \begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \quad (45)$$

$$Q_W = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 100 & 0 & 0 \\ 0 & 0 & 0 & 0 & 100 & 0 \\ 0 & 0 & 0 & 0 & 0 & 100 \end{pmatrix} \quad (46)$$

$$R_W = \begin{pmatrix} .0001 & 0 & 0 \\ 0 & .0001 & 0 \\ 0 & 0 & .0001 \end{pmatrix} \quad (47)$$

For Q_T and Q_W, there was a large penalty associated with body rate velocity. This limited the amount of overshoot experienced. The R_W matrix was down-weighted to 0.0001 to allow maximum used of SIMSAT's wheels—already tuned to a low level of acceleration. Keep in mind, this is only an arbitrary example. The response can be shaped, to a point, via weighting the Q and R matrices appropriately.

A relationship between the state thruster weighting matrix and the thruster rate gain was determined (Figure 50) for a specific case when all other factors were held constant. The relationship was determined to be linear (Equation 48) between the SDRE weight on the rate state and the PD rate gain.

$$K_{d,yaw} = 0.0457\, Q_{T,6,6} + 8.2456 \quad (48)$$

This relationship yielded approximately equal performance on the computer modeled SIMSAT, while thrust control was active, for single-axis control (yaw). No relationship was established between wheels gains and weighting matrices.

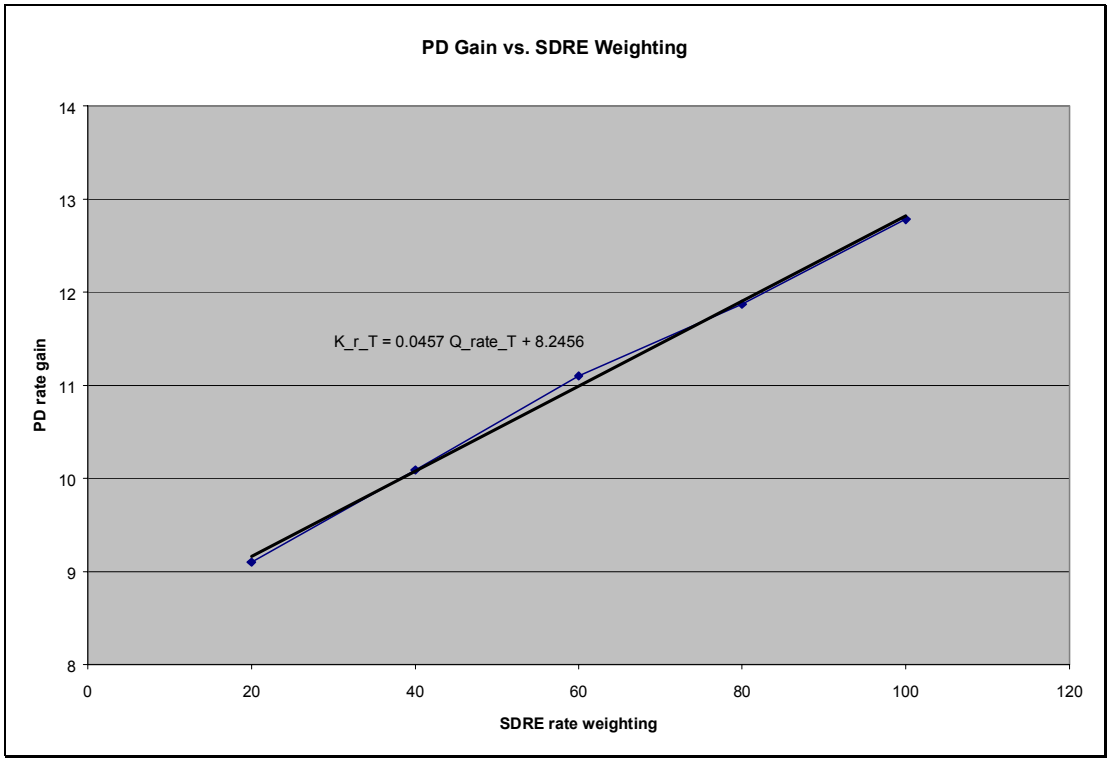


Figure 50. Limited relationship between PD rate gain and SDRE Q matrix

The two control methods seem to be equivalent in terms of thrust performance about a single axis. That is, it was possible to manipulate the rate gain and the yaw-axis weighting factor such that the performance was identical.

SIMSAT Computer Model Animation

For the purposes of visualization, an animation program was created to allow the model to be visualized (Figure 51). This program created a notional image of SIMSAT, and provided indicators of control input activation. In the figure shown, three pairs of

thrusters are shown firing. This file is capable of creating movie files from the animations.

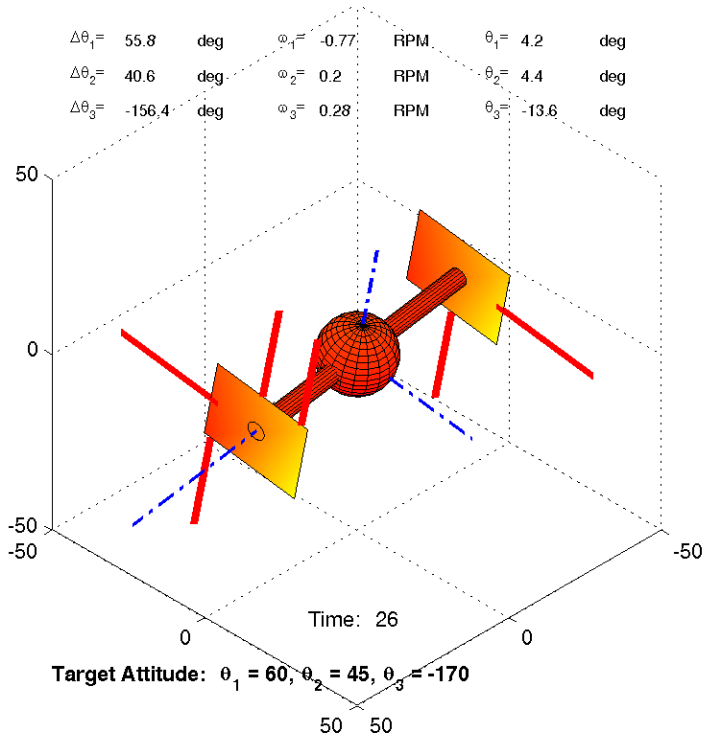


Figure 51. SIMSAT computer model animation screen shot.

V. Conclusions and Recommendations

Conclusions

There were several objectives to this thesis. They were as follows.

1. Design a satellite attitude control algorithm to provide near-optimal control to a satellite undergoing a large, multi-axis slewing maneuver, given full state feedback.
2. Implement this controller on SIMSAT using SIMSAT's reaction wheels as the only control input.
3. Design a thruster system and install it on SIMSAT.
4. Modify the controller designed for objective two, to allow near optimal control of SIMSAT with both thruster and reaction wheel control inputs.

The satellite attitude control algorithm was developed, and proven to work on a model similar to SIMSAT. This objective was completed.

The second objective was not achieved. It could be achieved if three conditions were met. First, a solution must be found to SIMSAT's long-axis sagging problem. Steps were taken during this thesis to procure equipment to potentially rigidize SIMSAT via a system of tension wires and turnbuckles. Second, the rapid drift of the pitch rate gyro must be eliminated. Third, the software issue restricting the Autobox from running an LQR routine must be fixed. Given these conditions, the second objective could be achieved. Note that if only single axis control is desired using LQR control, only the third impediment needs to be removed.

A thruster system was designed and installed on SIMSAT. It was effective in controlling SIMSAT's attitude, both by itself, and through combined control with the reaction wheels. This objective was completed.

Finally, if the step were taken to complete objective two, objective four could also be accomplished.

Recommendations

During the execution of the thesis and writing of this document, several issues have been noted that, if accomplished, would enhance SIMSAT and its use. These "action items" are listed below in order of importance, from the author's perspective. Following that, a recommendation is made on the general direction of study.

Action Items

First, solve the software problems preventing the SDRE controller from being implemented on SIMSAT. While it will not make much if any difference for single axis control, it will be a very good method of control once three-axis maneuvering is more feasible.

Second, it is recommended that AFIT/ENY obtain a high-pressure nitrogen fill station and tank storage or procure additional tanks and manufacture or purchase a carrying case for them. These tanks are susceptible to damage and dirt intrusion that can lead to failure. If a nitrogen fill station were obtained and a proper storage area found, a constant supply of nitrogen might be maintained and the bottles protected—no further action needed. If the intention is to continue outsourcing nitrogen fills, it would be desirable to obtain at least two more tanks and manufacture or procure a foam-lined case for their storage and transport. Also, future users of the system should be briefed on

some simple precautions to take with the bottles—ensuring the fill valves stay covered being the most pressing (that is how most bottles eventually become unserviceable according to Tim Hart of I-70 Paintball).

Third, rigidize SIMSAT. The limitation of control to one axis limits the experimental value of data attained from SIMSAT. This goal might be achievable via the use of tension wires.

Fourth, replace the rate gyro. The gyro used on SIMSAT is noisy and tends to drift unpredictably. In the interim, implement a low-pass filter to reduce the effects of noise.

Fifth, replace the nitrogen regulator. This regulator had difficulty maintaining a constant output, and had to be adjusted between runs to properly pressurize the thruster system.

Sixth, improve the computer reaction wheel model. While this remained a low priority in this thesis, the model matching benefit would be good for other applications.

Seventh, replace the solenoids and add a second nitrogen bottle. The solenoids purchased for SIMSAT work, but they are the reason the sustained thrust levels are low. The orifice of the solenoids in use is 1 mm in diameter—choking the flow. A larger orifice size is desirable. Along with this recommendation is the recommendation to add a second nitrogen bottle. Higher flow rates will mean higher performance demands and a rapidly emptied bottle. Currently, as the bottle empties, the weight of the expelled nitrogen is sufficient to put SIMSAT out of balance about the pitch axis. With two bottles paired to provide pressure and emptying at approximately the same rate, near balance might be maintained throughout test runs.

Future Study

The ideal fruition of this thesis would have been to obtain full three-axis control of SIMSAT using the SDRE controller. It is recommended that this goal be further pursued. While the SDRE controller does not appear to offer an advantage over the PD controller for the single-axis case in terms of thrust performance, it would outperform it for multiple axis control. Should the action items listed above be accomplished, SIMSAT may be more fully utilized as a test bed and a teaching tool.

Appendix A: Simulink Models

lqr_dual_sim.mdl

lqr_dual_sim.mdl/Dual Controller

lqr_dual_sim.mdl/Dual Controller/Thrust Controller

lqr_dual_sim.mdl/Dual Controller/Thrust Controller/T_LQR

lqr_dual_sim.mdl/Dual Controller/Thrust Controller/T_LQR/SDRE Solver

lqr_dual_sim.mdl/Dual Controller/Thrust Controller/T_LQR/SDRE Solver/Subsystem

lqr_dual_sim.mdl/Dual Controller/Wheel Controller

lqr_dual_sim.mdl/Plant Model

lqr_dual_sim.mdl/Plant Model/Wheel Dynamics

lqr_dual_sim.mdl/Plant Model/V to T

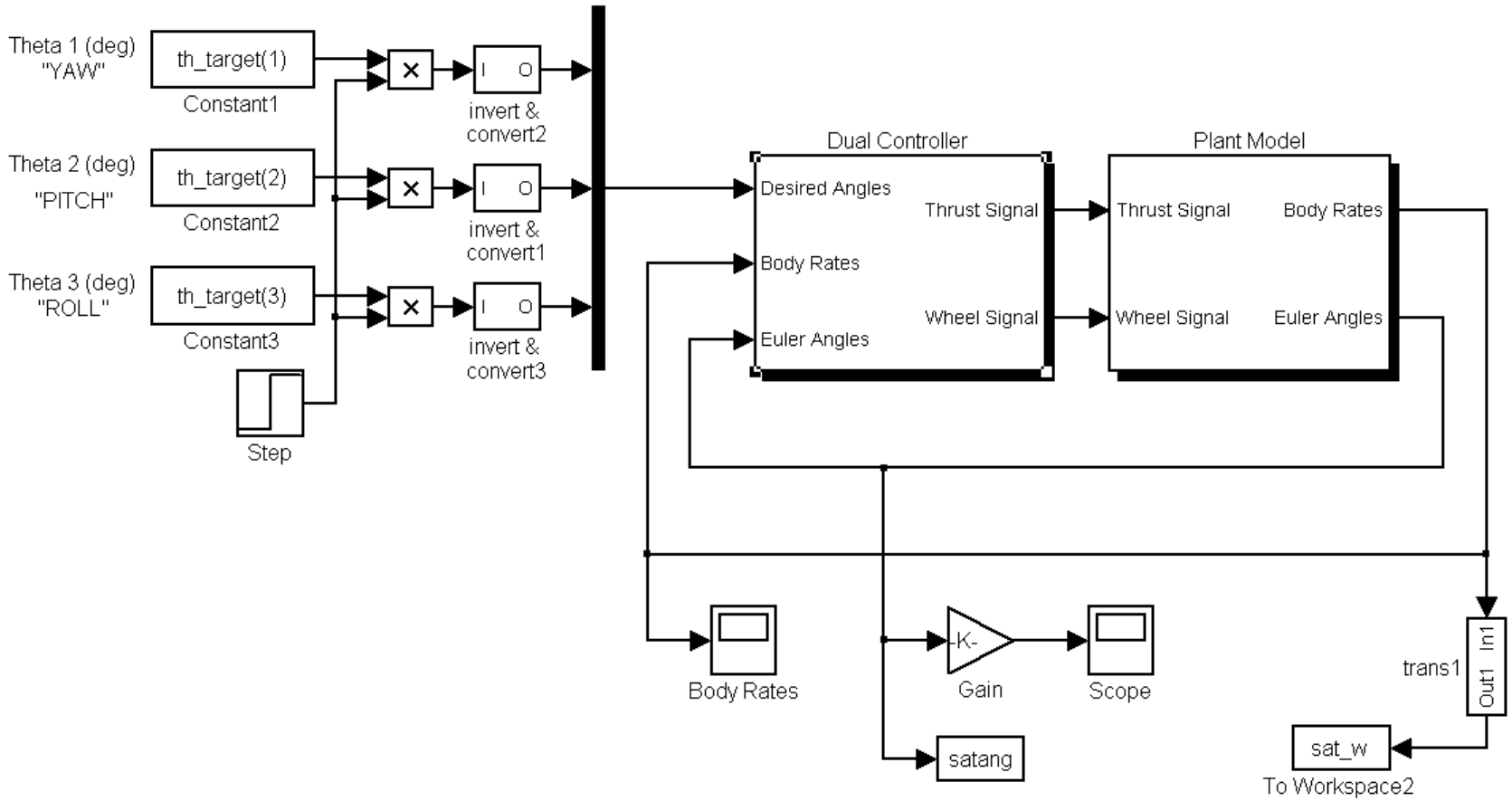
lqr_dual_sim.mdl/Plant Model/Dynamics

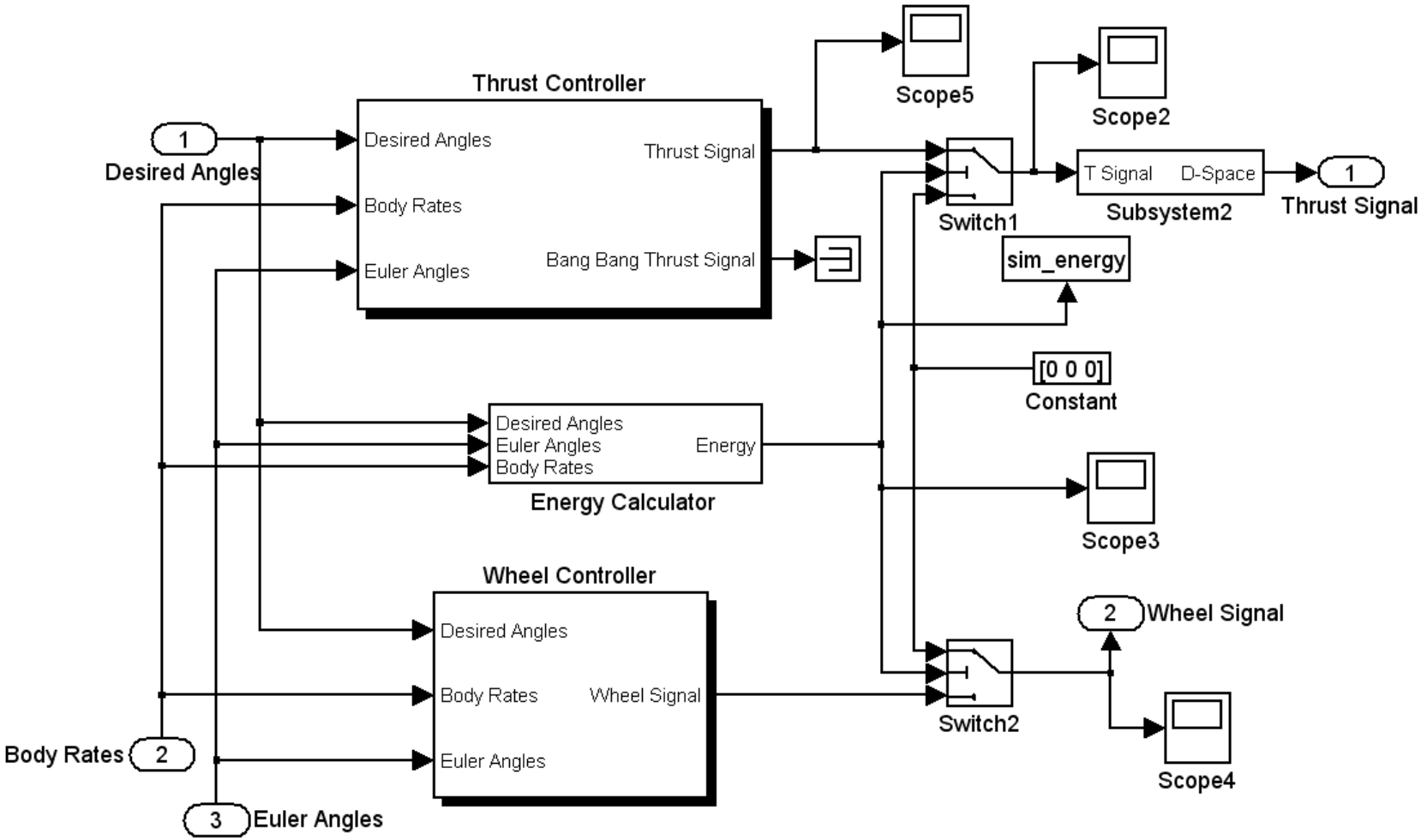
lqr_dual_sim.mdl/Plant Model/Euler 3-2-1 EOM3

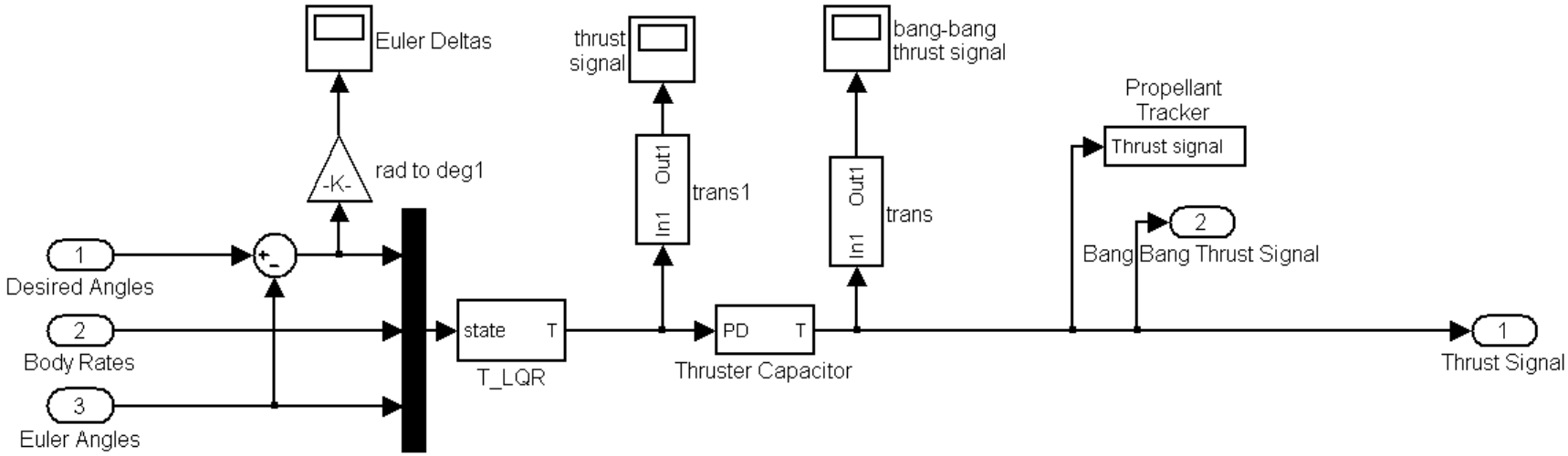
PD_dual2.mdl

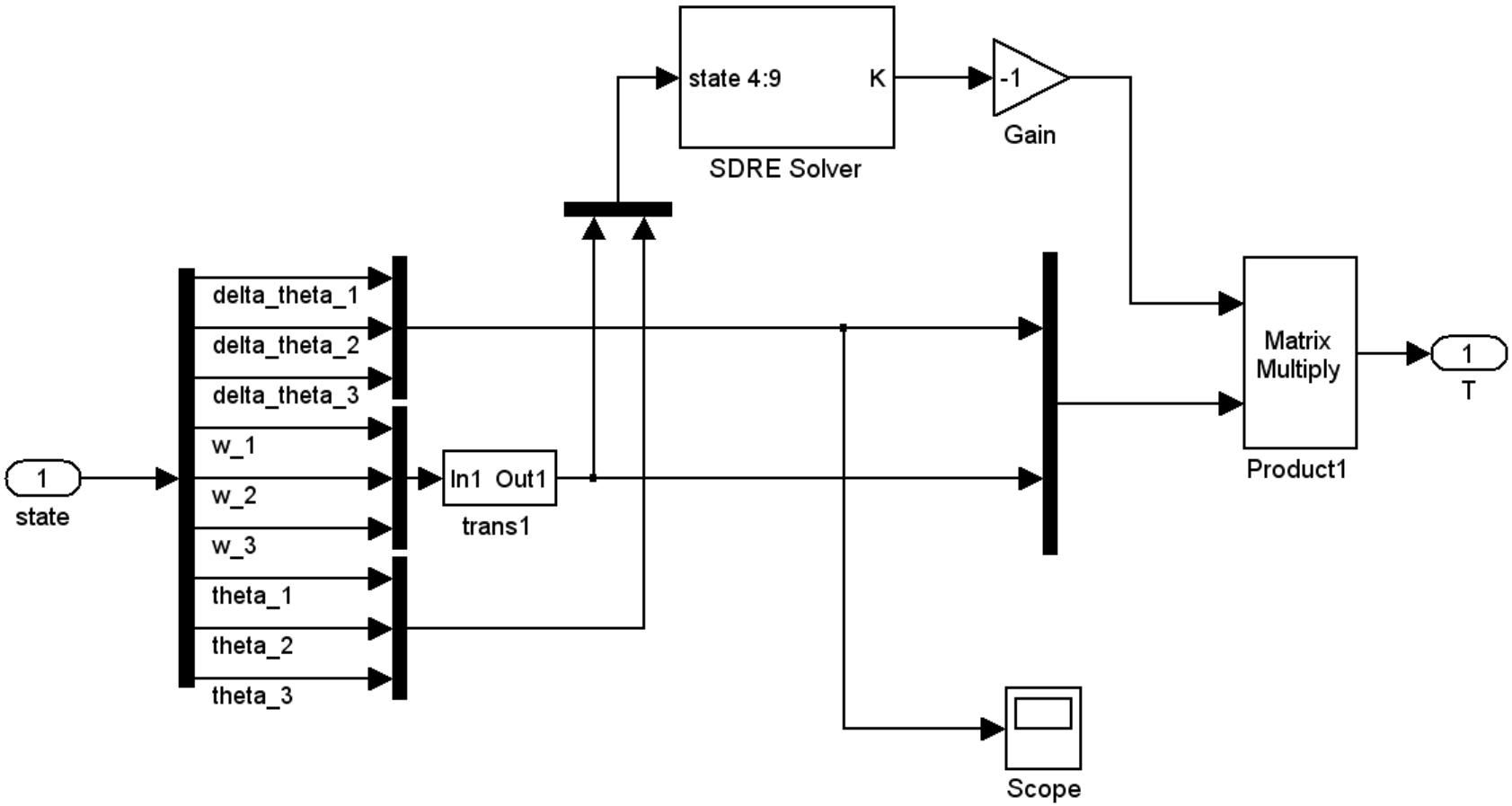
PD_dual2.mdl/Dual Controller

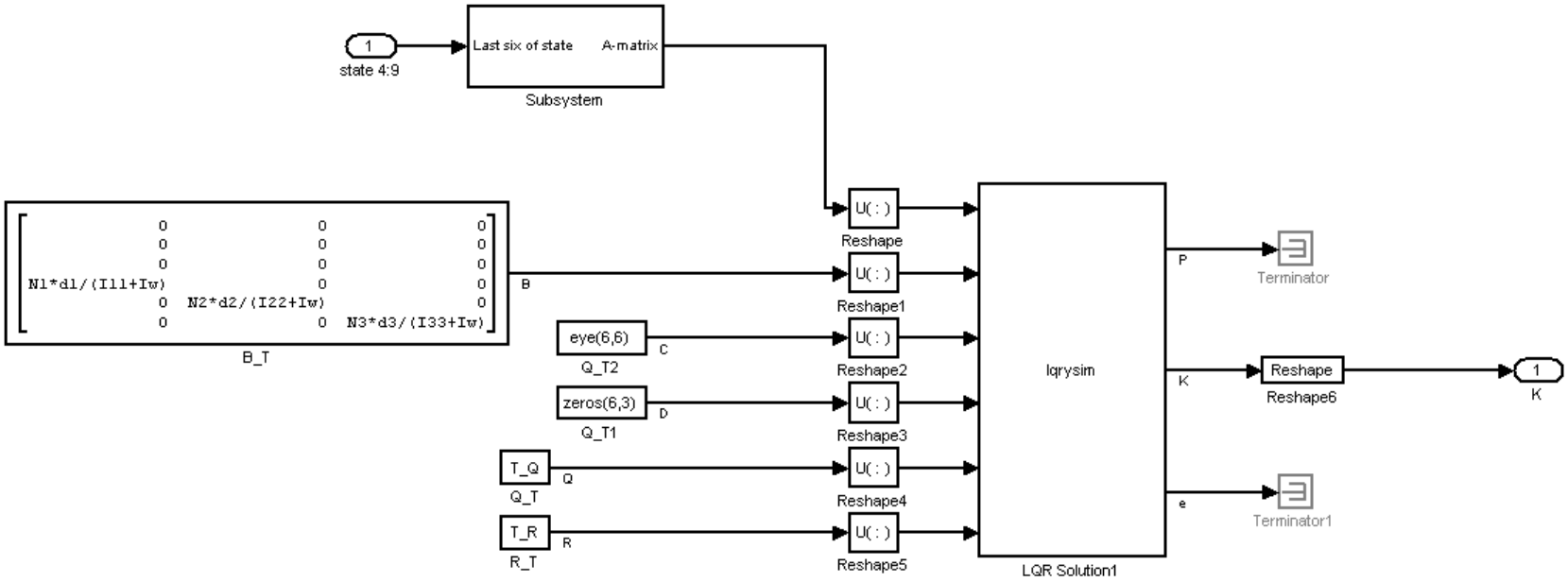
PD_dual2.mdl/Reaction Wheel Communication

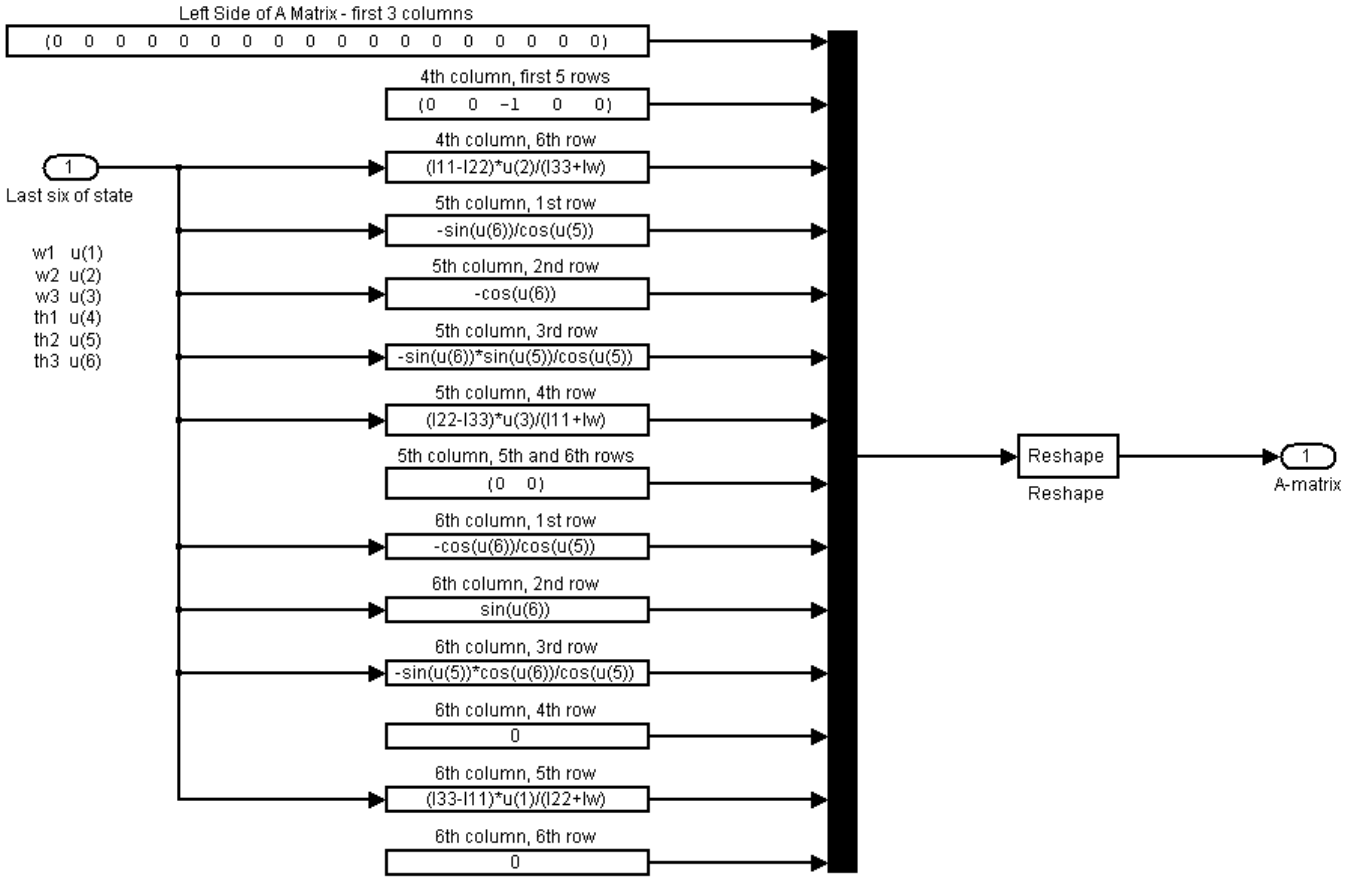

Theta 1 (deg)
"YAW"
th_target(1)
Constant1
Theta 2 (deg)
"PITCH"
th_target(2)
Constant2
Theta 3 (deg)
"ROLL"
th_target(3)
Constant3
Step
I O
invert &
convert2
I O
invert &
convert1
I O
invert &
convert3
Dual Controller
Desired Angles
Body Rates
Euler Angles
Thrust Signal
Wheel Signal
Plant Model
Thrust Signal
Wheel Signal
Body Rates
Euler Angles
Body Rates
K-
Gain
Scope
satang
trans1
In1 Out1
sat_w
To Workspace2

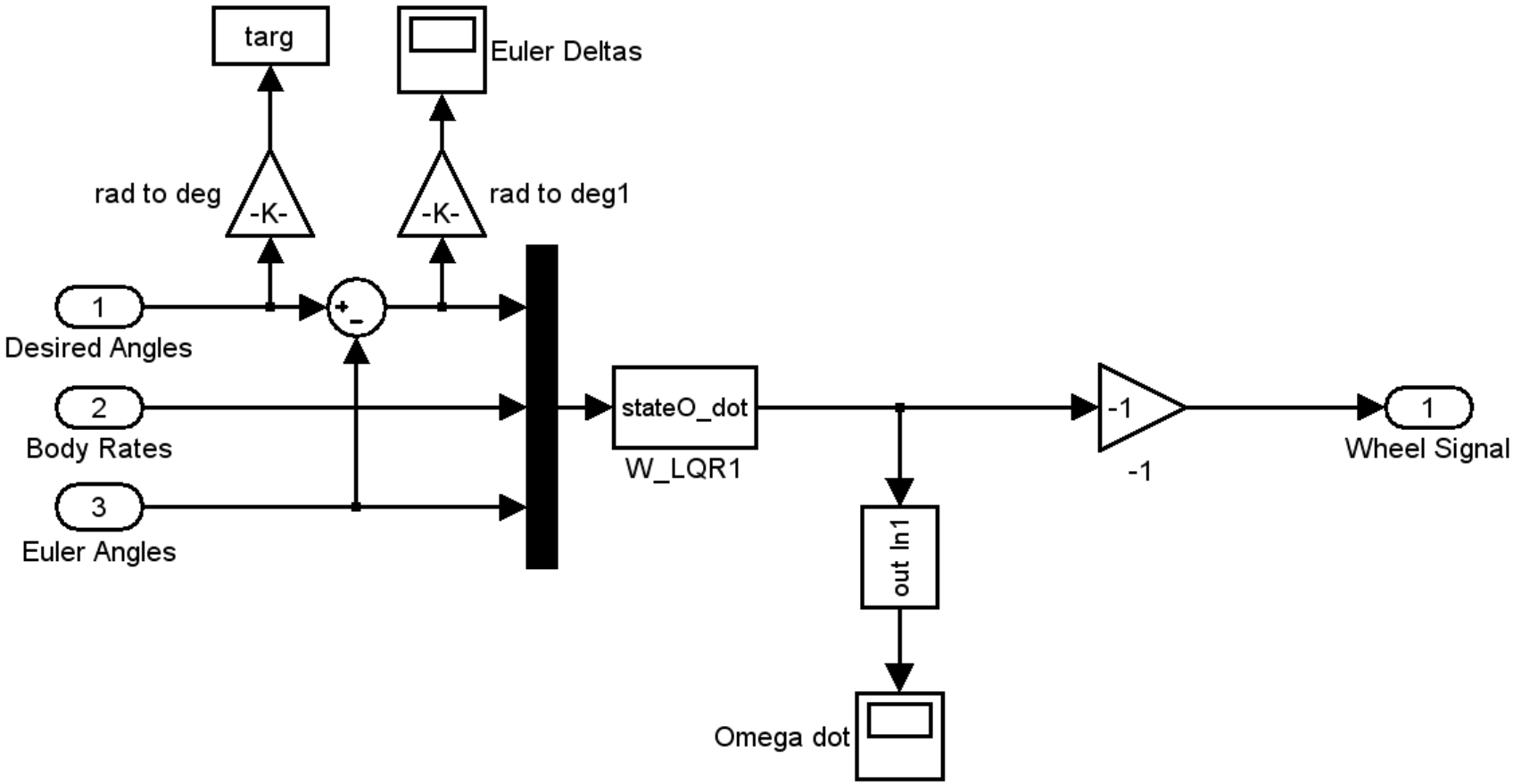

Thrust Controller
Desired Angles
Body Rates
Euler Angles
Thrust Signal
Bang Bang Thrust Signal
1
Desired Angles
Scope5
Scope2
Switch1
T Signal
D-Space
Subsystem2
1
Thrust Signal
sim_energy
[0 0 0]
Constant
Desired Angles
Euler Angles
Body Rates
Energy
Energy Calculator
Scope3
Wheel Controller
Desired Angles
Body Rates
Euler Angles
Wheel Signal
2
Wheel Signal
Switch2
Scope4
Body Rates
2
3
Euler Angles

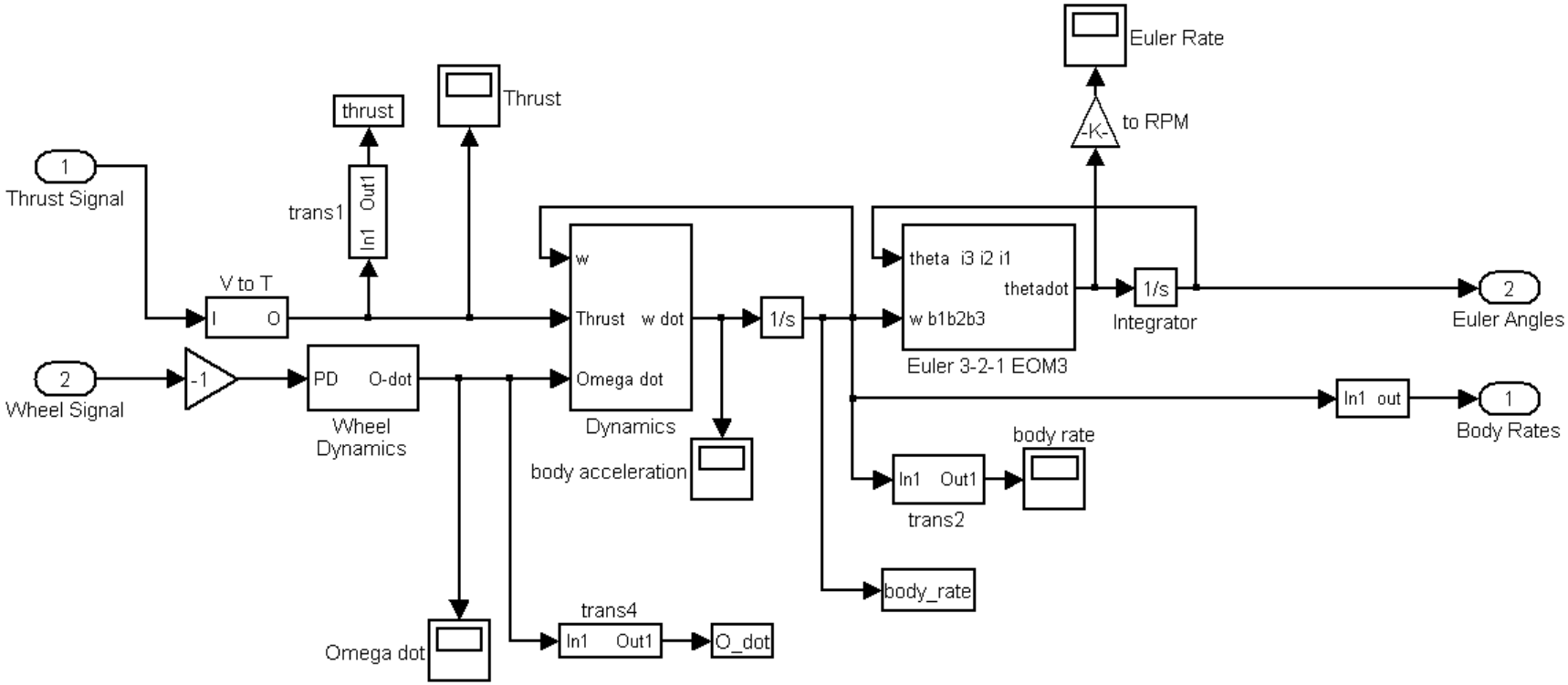

Euler Deltas
thrust signal
bang-bang thrust signal
Propellant Tracker
Thrust signal
-K-
rad to deg1
In1 Out1
trans1
In1 Out1
trans
1
Desired Angles
2
Body Rates
3
Euler Angles
state T
T_LQR
PD T
Thruster Capacitor
2
Bang Bang Thrust Signal
1
Thrust Signal

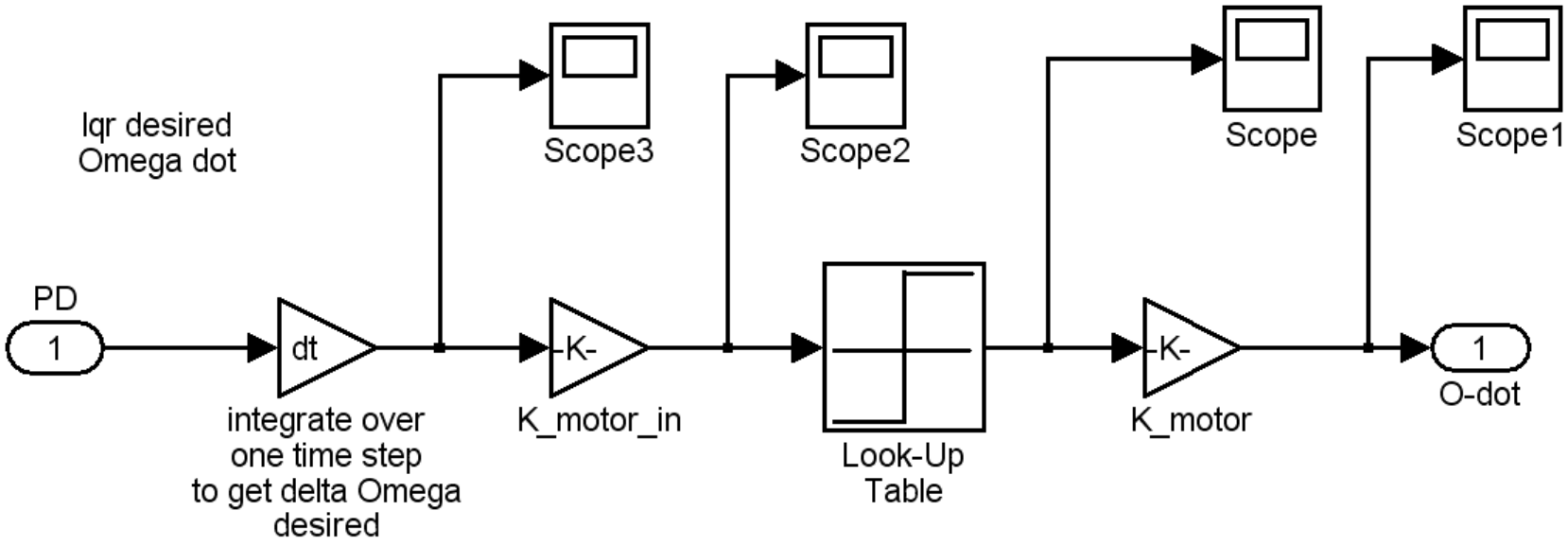

state 4:9
K
SDRE Solver
-1
Gain
delta_theta_1
delta_theta_2
delta_theta_3
w_1
w_2
w_3
theta_1
theta_2
theta_3
1
state
In1 Out1
trans1
Matrix
Multiply
Product1
1
T
Scope

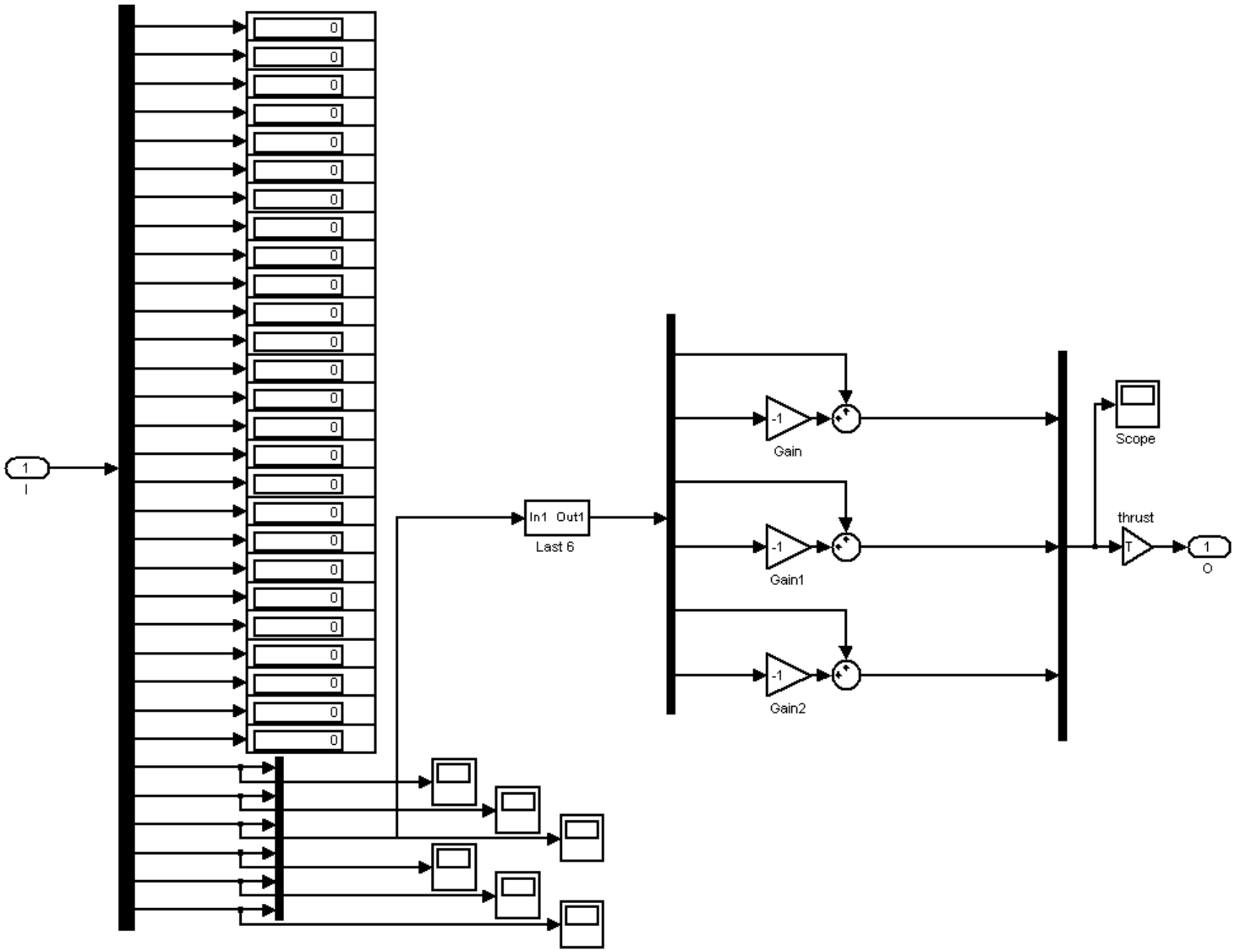

1
state 4:9
Last six of state
A-matrix
Subsystem
0 0 0
0 0 0
0 0 0
N1*d1/(I11+Iw) 0 0
0 N2*d2/(I22+Iw) 0
0 0 N3*d3/(I33+Iw)
B_T
B
eye(6,6)
C
Q_T2
zeros(6,3)
D
Q_T1
T_Q
Q
Q_T
T_R
R
R_T
U(:)
Reshape
Reshape1
Reshape2
Reshape3
Reshape4
Reshape5
lqrysim
LQR Solution1
P
Terminator
K
Reshape
Reshape6
1
K
e
Terminator1

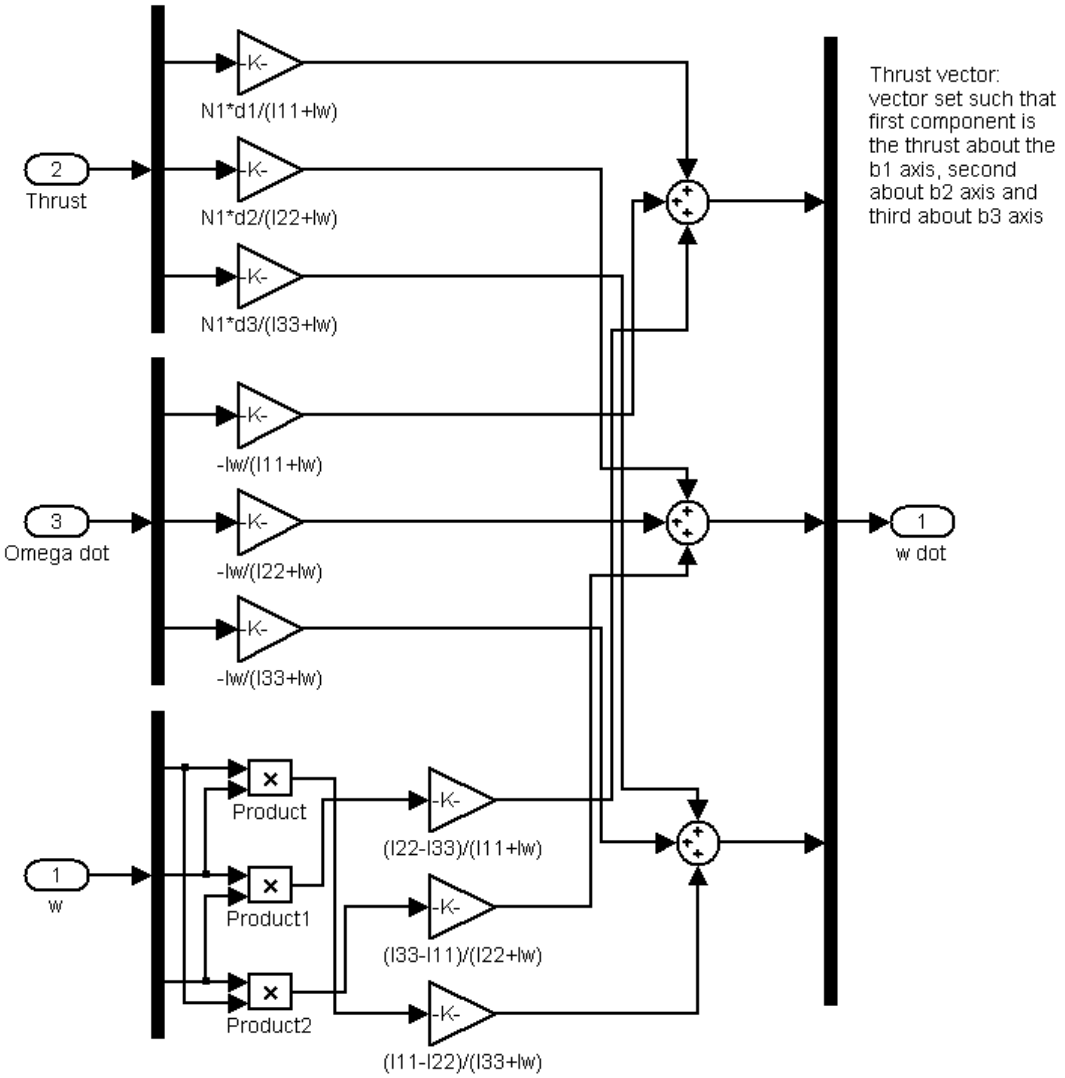

Left Side of A Matrix - first 3 columns
(0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0)
4th column, first 5 rows
(0 0 -1 0 0)
4th column, 6th row
(I11-I22)*u(2)/(I33+Iw)
5th column, 1st row
-sin(u(6))/cos(u(5))
5th column, 2nd row
-cos(u(6))
5th column, 3rd row
-sin(u(6))*sin(u(5))/cos(u(5))
5th column, 4th row
(I22-I33)*u(3)/(I11+Iw)
5th column, 5th and 6th rows
(0 0)
6th column, 1st row
-cos(u(6))/cos(u(5))
6th column, 2nd row
sin(u(6))
6th column, 3rd row
-sin(u(5))*cos(u(6))/cos(u(5))
6th column, 4th row
0
6th column, 5th row
(I33-I11)*u(1)/(I22+Iw)
6th column, 6th row
0
1
Last six of state
w1 u(1)
w2 u(2)
w3 u(3)
th1 u(4)
th2 u(5)
th3 u(6)
Reshape
Reshape
1
A-matrix

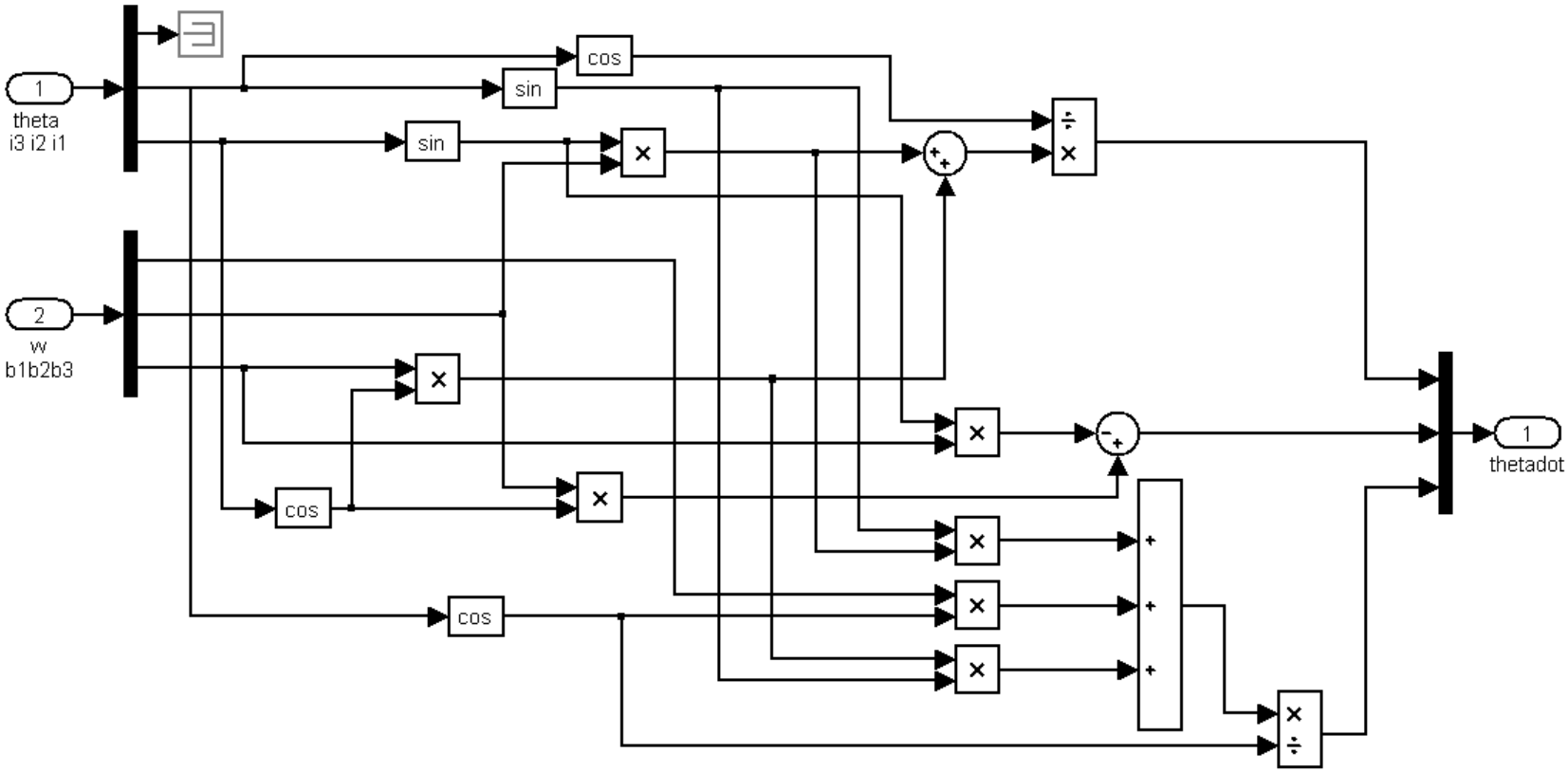

targ
Euler Deltas
rad to deg
-K-
-K-
rad to deg1
1
Desired Angles
2
Body Rates
3
Euler Angles
stateO_dot
W_LQR1
-1
-1
1
Wheel Signal
out In1
Omega dot


Euler Rate
to RPM
-K-
thrust
Thrust
1
Thrust Signal
trans1
In1 Out1
V to T
I O
w
Thrust w dot
Omega dot
Dynamics
1/s
theta i3 i2 i1
thetadot
w b1b2b3
Euler 3-2-1 EOM3
1/s
Integrator
2
Euler Angles
2
Wheel Signal
-1
PD O-dot
Wheel
Dynamics
body acceleration
In1 out
1
Body Rates
body rate
In1 Out1
trans2
body_rate
trans4
In1 Out1
O_dot
Omega dot


lqr desired
Omega dot
Scope3
Scope2
Scope
Scope1
PD
1
dt
-K-
-K-
1
integrate over
one time step
to get delta Omega
desired
K_motor_in
Look-Up
Table
K_motor
O-dot


1
I
0
In1 Out1
Last 6
-1
Gain
-1
Gain1
-1
Gain2
Scope
thrust
T
1
O

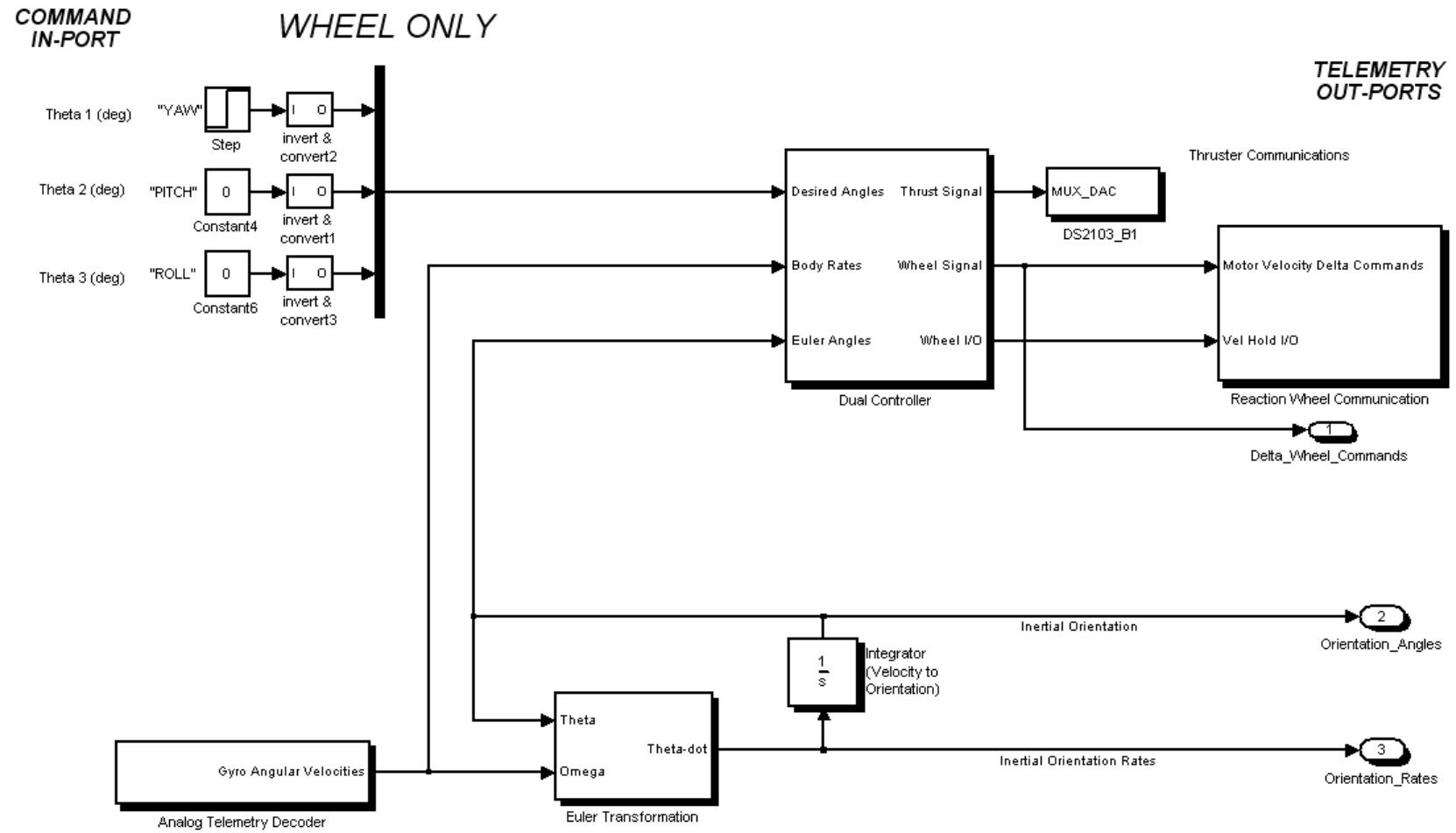

2
Thrust
K-
N1*d1/(I11+Iw)
K-
N1*d2/(I22+Iw)
K-
N1*d3/(I33+Iw)
3
Omega dot
K-
-Iw/(I11+Iw)
K-
-Iw/(I22+Iw)
K-
-Iw/(I33+Iw)
1
w
Product
Product1
Product2
K-
(I22-I33)/(I11+Iw)
K-
(I33-I11)/(I22+Iw)
K-
(I11-I22)/(I33+Iw)
Thrust vector:
vector set such that
first component is
the thrust about the
b1 axis, second
about b2 axis and
third about b3 axis
1
w dot

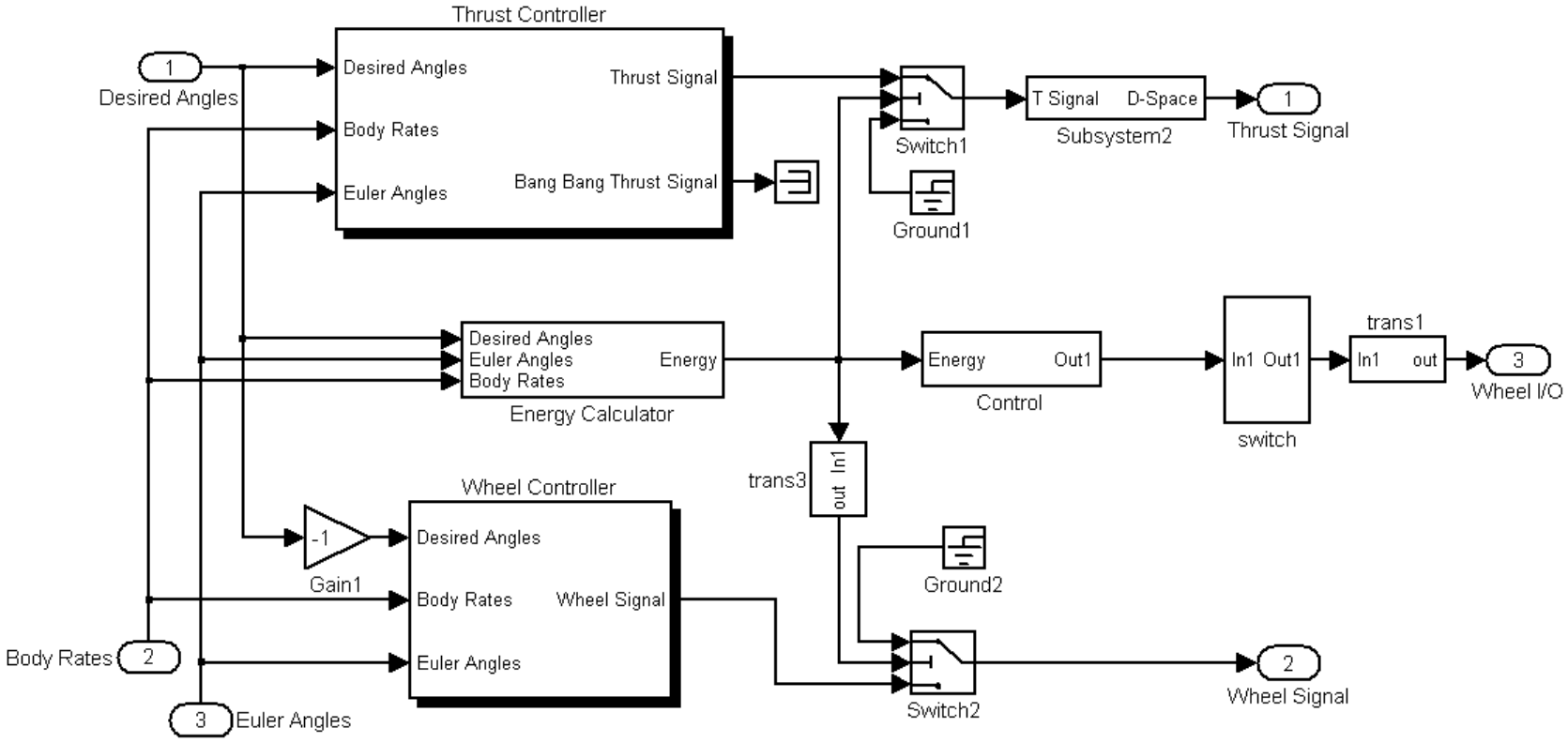

1
theta
i3 i2 i1
2
w
b1b2b3
cos
sin
sin
cos
cos
1
thetadot

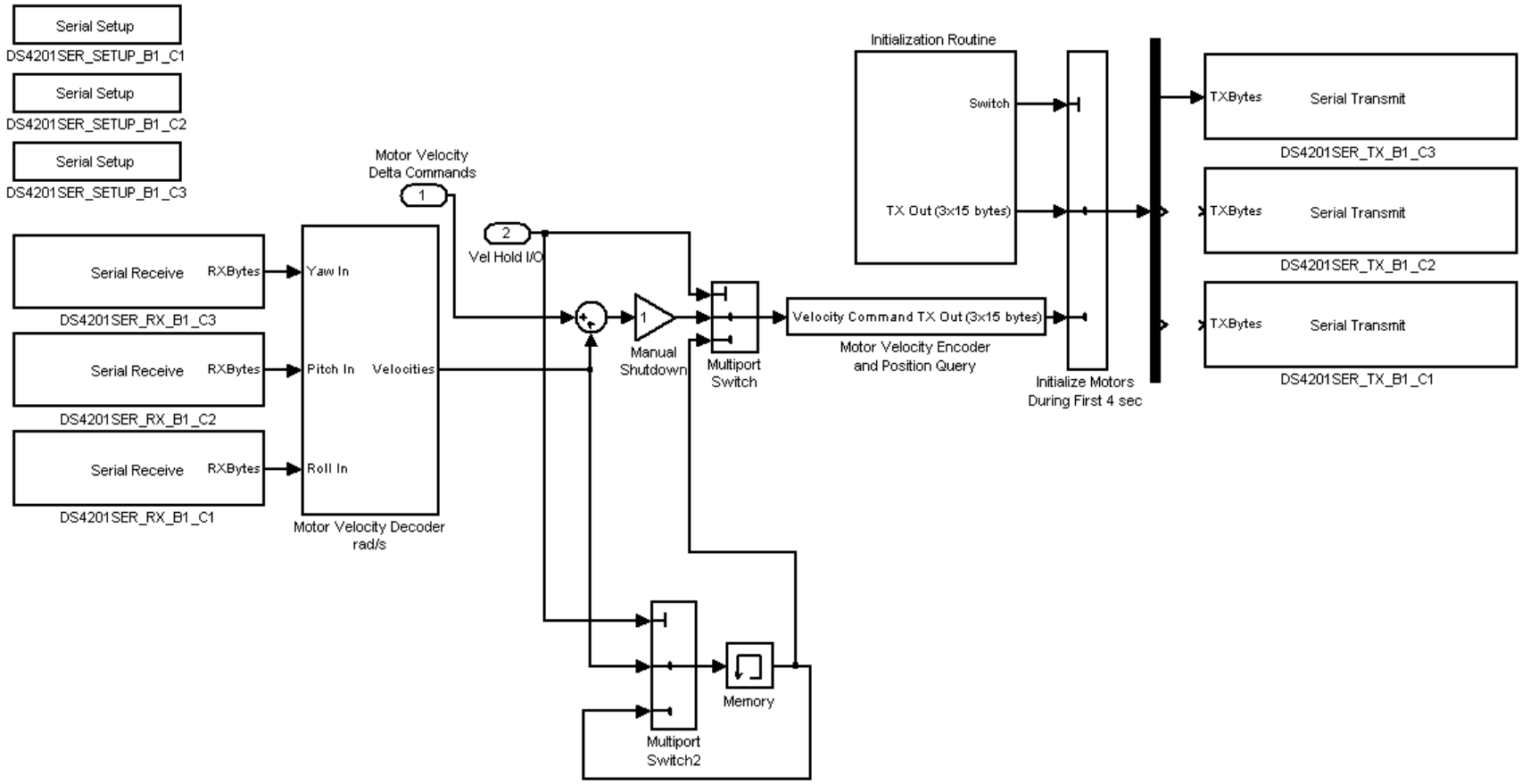
RTI Data

FRENCHNIK TOP LEVEL SOFTWARE ARCHITECTURE

NOTE: Euler 3-2-1 Angles used to describe Yaw, Pitch, Roll (respectively)
All signals are in the order of the body 3,2,1 axes (Yaw, Pitch, Roll respectively)
All angles are in radians, all rates are in radians/second




Thrust Controller
1
Desired Angles
Desired Angles
Body Rates
Euler Angles
Thrust Signal
Bang Bang Thrust Signal
Switch1
Ground1
T Signal
D-Space
Subsystem2
1
Thrust Signal
Desired Angles
Euler Angles
Body Rates
Energy
Energy Calculator
Energy
Out1
Control
In1 Out1
switch
trans1
In1
out
3
Wheel I/O
trans3
out In1
Wheel Controller
-1
Gain1
Desired Angles
Body Rates
Euler Angles
Wheel Signal
Ground2
Body Rates
2
3
Euler Angles
Switch2
2
Wheel Signal


Serial Setup
DS4201SER_SETUP_B1_C1
Serial Setup
DS4201SER_SETUP_B1_C2
Serial Setup
DS4201SER_SETUP_B1_C3
Motor Velocity
Delta Commands
1
2
Vel Hold I/O
Serial Receive
RXBytes
DS4201SER_RX_B1_C3
Serial Receive
RXBytes
DS4201SER_RX_B1_C2
Serial Receive
RXBytes
DS4201SER_RX_B1_C1
Yaw In
Pitch In
Roll In
Velocities
Motor Velocity Decoder
rad/s
1
Manual
Shutdown
Multiport
Switch
Memory
Multiport
Switch2
Initialization Routine
Switch
TX Out (3x15 bytes)
Velocity Command TX Out (3x15 bytes)
Motor Velocity Encoder
and Position Query
Initialize Motors
During First 4 sec
TXBytes
Serial Transmit
DS4201SER_TX_B1_C3
TXBytes
Serial Transmit
DS4201SER_TX_B1_C2
TXBytes
Serial Transmit
DS4201SER_TX_B1_C1

Appendix B: MATLAB® Code

simlaunch6_lqr_sim.m

simlaunch6.m

sat_anim_7.m

data_get.m

```
% simlaunch6_lqr_sim
% David French
% global I11 I22 I33 Iw N1 N2 N3 d1 d2 d3 A1 A2 A3 A4 A5 A6 B1 B2 B3 B4
B5 B6

clc
close all
clear all

global I11 I22 I33 Iw N1 N2 N3 d1 d2 d3 satnow

% ***********TARGET ATTITUDE*****************
th_target=[175 45 -90]'

% ************************************************
% *                                              *
% *                 CONTROL VARIABLES            *
% *                                              *
% ************************************************

T_Qu=1;
T_Ql=100;
T_Ra=1;

W_Qu=1;
W_Ql=100;
W_Ra=.0001;

ep_K_T=0.005    %thruster K "do nothing" limit

eswitch=.001
%control switching threshold

% ************************************************
% *                                              *
% *                 SYSTEM CONSTANTS             *
% *                                              *
% ************************************************

% Wheel motor gains
K_motor_in=.1
K_motor=1


T_av=.045

T_bias=1      %negative thrust stronger

T=T_av*(1+T_bias)/2


% ************************************************
% time
dt=.05       % time step
tend=300   % end time

satnow=[0 0 0]
th_1=0
th_2=0
th_3=0
```

```
w_1=0
w_2=0
w_3=0

% Wheel MOI
Iw=66.17/32.17

%Max wheel speed
Om_max=3400*2*pi/60

%Max wheel torque
Tq_max=760/16

% SIMSAT MOIs
I11=3800.66/32.2
I22=38318/32.2
I33=36652/32.2

% Thruster moment arms (in)
d1=12
d2=36
d3=36

% Number of thrusters per axis
N1=1
N2=1        % Internal Sum adds to 3 (4 Feb)
N3=1

%Energy Poly Coef
P2_P=10
P2_K=100

%Braking constant
K_brake=1

% Voltage to send to D-Space (volts)
V_on=1

% D-space relay on/off settings
R_on=1
R_off=0


% Wheel Control

W_Q1=[W_Qu 0 0  0 0 0];
W_Q2=[0 W_Qu 0  0 0 0];
W_Q3=[0 0 W_Qu  0 0 0];
W_Q4=[0 0 0  W_Ql 0 0];
W_Q5=[0 0 0  0 W_Ql 0];
W_Q6=[0 0 0  0 0 W_Ql];

W_R1=[W_Ra 0 0];
W_R2=[0 W_Ra 0];
W_R3=[0 0 W_Ra];

W_Q=[W_Q1;W_Q2;W_Q3;W_Q4;W_Q5;W_Q6];
W_R=[W_R1;W_R2;W_R3];

% Thruster Control
```

```
T_Q1=[T_Qu 0 0  0 0 0];
T_Q2=[0 T_Qu 0  0 0 0];
T_Q3=[0 0 T_Qu  0 0 0];
T_Q4=[0 0 0  T_Ql 0 0];
T_Q5=[0 0 0  0 T_Ql 0];
T_Q6=[0 0 0  0 0 T_Ql];

T_R1=[T_Ra 0 0];
T_R2=[0 T_Ra 0];
T_R3=[0 0 T_Ra];

T_Q=[T_Q1;T_Q2;T_Q3;T_Q4;T_Q5;T_Q6];
T_R=[T_R1;T_R2;T_R3];


% open sat_anim_7
sim lqr_dual_sim
% data_get
% data_sim_lqr
data_sim_lqr
```

```
% simlaunch6
% David French
% global I11 I22 I33 Iw N1 N2 N3 d1 d2 d3 A1 A2 A3 A4 A5 A6 B1 B2 B3 B4
B5 B6

clc
close all
clear all

global I11 I22 I33 Iw N1 N2 N3 d1 d2 d3 satnow

% ***********TARGET ATTITUDE*****************
th_target=40

% ************************************************
% *                                              *
% *                 CONTROL VARIABLES            *
% *                                              *
% ************************************************

ep_K_T=0.05    %thruster K "do nothing" limit

K_r_T=5.3   %thruster controller's rate gain
K_d_T=1      %thruster controller's delta gain
K_o_T=1      %thruster controller's overall gain

K_r_W=K_r_T*1000   %wheel controller's rate gain
K_d_W=K_d_T*1000   %wheel controller's delta gain
K_o_W=-1.1     %wheel controller's overall gain

eswitch=.015
%control switching threshhold

% ************************************************
% *                                              *
% *                 SYSTEM CONSTANTS             *
% *                                              *
% ************************************************

% Wheel motor gains
K_motor_in=.1
K_motor=1


T_av=.045

T_bias=1.25      %negative thrust stronger

T=T_av*(1+T_bias)/2


% ************************************************
% time
dt=.05       % time step
tend=120   % end time

satnow=[0 0 0]
th_1=0
th_2=0
th_3=0
```

```
w_1=0
w_2=0
w_3=0

% Wheel MOI
Iw=66.17/32.17

%Max wheel speed
Om_max=3400*2*pi/60

%Max wheel torque
Tq_max=760/16

% SIMSAT MOIs
I11=3800.66/32.2
I22=38318/32.2
I33=36652/32.2

% Thruster moment arms (in)
d1=12
d2=36
d3=36

% Number of thrusters per axis
N1=1
N2=1        % Internal Sum adds to 3 (4 Feb)
N3=1

%Energy Poly Coef
P2_P=10
P2_K=100

%Braking constant
K_brake=1

% Voltage to send to D-Space (volts)
V_on=1

% D-space relay on/off settings
R_on=1
R_off=0

% open sat_anim_7
sim PD_dual_sim
data_get
% data_sim2
```

```
% sat_anim_7

close all
figure
hold on
set(gca,'nextplot','replacechildren');

% Animation time increment
inc=20;

% wheel torque vector scaling
w_scl=20

tlen=30  %thrust vector length

% Initialize movie file
clear M

% Model specifications

bcom=20;    % arm and sphere detail
R_a=3;      % arm radius
alen=36;    % arm length

% end plate colors
co1=1;
co2=12;
co3=45;
co4=60;

R_s=12;     % sphere radius

s=size(satang); % size of angle array for animation

% map positive unit vectors
tipx=[60 0 0];
tipy=[0 36 0];
tipz=[0 0 36];


% thrust vectors initialize (relative to starting position)
% Axis 1
T1i_1p_a=[36 -12 12];
T1i_1p_b=[36 -12 12+tlen];

T1i_2p_a=[36 12 -12];
T1i_2p_b=[36 12 -12-tlen];

T1i_1n_a=[36 12 12];
T1i_1n_b=[36 12 12+tlen];

T1i_2n_a=[36 -12 -12];
T1i_2n_b=[36 -12 -12-tlen];

% Axis 2
T2i_1p_a=[36 0 12];
T2i_1p_b=[36 0 12+tlen];

T2i_2p_a=[-36 0 -12];
T2i_2p_b=[-36 0 -12-tlen];
```

```
T2i_1n_a=[-36 0 12];
T2i_1n_b=[-36 0 12+tlen];

T2i_2n_a=[36 0 -12];
T2i_2n_b=[36 0 -12-tlen];

% Axis 3
T3i_1p_a=[36 -12 0];
T3i_1p_b=[36 -12-tlen 0];

T3i_2p_a=[-36 12 0];
T3i_2p_b=[-36 12+tlen 0];

T3i_1n_a=[36 12 0];
T3i_1n_b=[36 12+tlen 0];

T3i_2n_a=[-36 -12 0];
T3i_2n_b=[-36 -12-tlen 0];

% map ends
pm1=[-36 -12 -12
    -36 -12 12
    -36 12 12
    -36 12 -12];

pm2=[36 -12 -12
    36 -12 12
    36 12 12
    36 12 -12];

% map sphere
[sx sy sz]=sphere(bcom);
% expand sphere to radius
sx=sx*R_s;
sy=sy*R_s;
sz=sz*R_s;

% map arms
[az ay ax]=cylinder(R_a,bcom);
% lengthen
ax=ax*alen;

% Assemble sphere and arms
bx=[sx;ax;-ax];
by=[sy;ay;-ay];
bz=[sz;az;-az];

% sphere and arm map data size for rotation routine
siz=size(bx);
m=siz(1);
n=siz(2);

% choose colormap
colormap('autumn')
% colormap('flag')
% colormap('hot')
% colormap('pink')
% colormap('prism')
% colormap('cool')
% colormap('copper')
```

```
% animation sequence
f=0;    %initialize frame number
for nn=1:inc:s(1);
    f=f+1;
    t=nn*dt;
    th1=satang(nn,1);
    th2=satang(nn,2);
    th3=satang(nn,3);
    w1=sat_w(nn,1);
    w2=sat_w(nn,2);
    w3=sat_w(nn,3);

    % Rotation Matrix
    ROT1=[cos(th1)*cos(th2) sin(th1)*cos(th2) -sin(th2)];
    ROT2=[cos(th1)*sin(th2)*sin(th3)-sin(th1)*cos(th3)
cos(th1)*cos(th3)+sin(th1)*sin(th2)*sin(th3) cos(th2)*sin(th3)];
    ROT3=[sin(th1)*sin(th3)+cos(th1)*sin(th2)*cos(th3) -
cos(th1)*sin(th3)+sin(th1)*sin(th2)*cos(th3) cos(th2)*cos(th3)];

    ROT=[ROT1;ROT2;ROT3];

    % Positive unit vector rotations
    tip1=tipx*ROT;
    tip2=tipy*ROT;
    tip3=tipz*ROT;

    % patch rotation
    p1=pm1*ROT;
    p2=pm2*ROT;

    % body rotation
    for i=1:m;
        for j=1:n;
            v=[bx(i,j) by(i,j) bz(i,j)];
            bv=v*ROT;
            b1(i,j)=bv(1);
            b2(i,j)=bv(2);
            b3(i,j)=bv(3);
        end
    end

    %animate

    grid on
    surf(b1,b2,b3)

    % ends
    patch(p1(:,1),p1(:,2),p1(:,3),[co1 co2 co3 co4])
    patch(p2(:,1),p2(:,2),p2(:,3),[co1 co2 co3 co4])

    % unit vectors
    line([0 tip1(1)],[0 tip1(2)],[0 tip1(3)],'LineStyle','-
.','LineWidth',2)
    line([0 tip2(1)],[0 tip2(2)],[0 tip2(3)],'LineStyle','-
.','LineWidth',2)
    line([0 tip3(1)],[0 tip3(2)],[0 tip3(3)],'LineStyle','-
.','LineWidth',2)

    % thrust vector animation
    % Axis 1
    if thrust(nn,3)>0
```

```
        T1b_1p_a=T1i_1p_a*ROT;
        T1b_1p_b=T1i_1p_b*ROT;
        T1b_2p_a=T1i_2p_a*ROT;
        T1b_2p_b=T1i_2p_b*ROT;
        line([T1b_1p_a(1) T1b_1p_b(1)],[T1b_1p_a(2)
T1b_1p_b(2)],[T1b_1p_a(3) T1b_1p_b(3)],'Color','r','LineWidth',4)
        line([T1b_2p_a(1) T1b_2p_b(1)],[T1b_2p_a(2)
T1b_2p_b(2)],[T1b_2p_a(3) T1b_2p_b(3)],'Color','r','LineWidth',4)
    end

    if thrust(nn,3)<0
        T1b_1n_a=T1i_1n_a*ROT;
        T1b_1n_b=T1i_1n_b*ROT;
        T1b_2n_a=T1i_2n_a*ROT;
        T1b_2n_b=T1i_2n_b*ROT;
        line([T1b_1n_a(1) T1b_1n_b(1)],[T1b_1n_a(2)
T1b_1n_b(2)],[T1b_1n_a(3) T1b_1n_b(3)],'Color','r','LineWidth',4)
        line([T1b_2n_a(1) T1b_2n_b(1)],[T1b_2n_a(2)
T1b_2n_b(2)],[T1b_2n_a(3) T1b_2n_b(3)],'Color','r','LineWidth',4)
    end

    % Axis 2
    if thrust(nn,2)>0
        T2b_1p_a=T2i_1p_a*ROT;
        T2b_1p_b=T2i_1p_b*ROT;
        T2b_2p_a=T2i_2p_a*ROT;
        T2b_2p_b=T2i_2p_b*ROT;
        line([T2b_1p_a(1) T2b_1p_b(1)],[T2b_1p_a(2)
T2b_1p_b(2)],[T2b_1p_a(3) T2b_1p_b(3)],'Color','r','LineWidth',4)
        line([T2b_2p_a(1) T2b_2p_b(1)],[T2b_2p_a(2)
T2b_2p_b(2)],[T2b_2p_a(3) T2b_2p_b(3)],'Color','r','LineWidth',4)
    end

    if thrust(nn,2)<0
        T2b_1n_a=T2i_1n_a*ROT;
        T2b_1n_b=T2i_1n_b*ROT;
        T2b_2n_a=T2i_2n_a*ROT;
        T2b_2n_b=T2i_2n_b*ROT;
        line([T2b_1n_a(1) T2b_1n_b(1)],[T2b_1n_a(2)
T2b_1n_b(2)],[T2b_1n_a(3) T2b_1n_b(3)],'Color','r','LineWidth',4)
        line([T2b_2n_a(1) T2b_2n_b(1)],[T2b_2n_a(2)
T2b_2n_b(2)],[T2b_2n_a(3) T2b_2n_b(3)],'Color','r','LineWidth',4)
    end

    % Axis 3
    if thrust(nn,1)>0
        T3b_1p_a=T3i_1p_a*ROT;
        T3b_1p_b=T3i_1p_b*ROT;
        T3b_2p_a=T3i_2p_a*ROT;
        T3b_2p_b=T3i_2p_b*ROT;
        line([T3b_1p_a(1) T3b_1p_b(1)],[T3b_1p_a(2)
T3b_1p_b(2)],[T3b_1p_a(3) T3b_1p_b(3)],'Color','r','LineWidth',4)
        line([T3b_2p_a(1) T3b_2p_b(1)],[T3b_2p_a(2)
T3b_2p_b(2)],[T3b_2p_a(3) T3b_2p_b(3)],'Color','r','LineWidth',4)
    end

    if thrust(nn,1)<0
        T3b_1n_a=T3i_1n_a*ROT;
        T3b_1n_b=T3i_1n_b*ROT;
        T3b_2n_a=T3i_2n_a*ROT;
        T3b_2n_b=T3i_2n_b*ROT;
```

```
        line([T3b_1n_a(1) T3b_1n_b(1)],[T3b_1n_a(2)
T3b_1n_b(2)],[T3b_1n_a(3) T3b_1n_b(3)],'Color','r','LineWidth',4)
        line([T3b_2n_a(1) T3b_2n_b(1)],[T3b_2n_a(2)
T3b_2n_b(2)],[T3b_2n_a(3) T3b_2n_b(3)],'Color','r','LineWidth',4)
    end

    % WHEEL ANIMATION
    % (wheel) (axis) (frame)_(wheel dir indicator #)_(pos/neg)_(end)

    % Axis 1
    if O_dot(nn,3)<0
        w1len=abs(O_dot(nn,1))*w_scl;

        W1i_1p_a=[37 -13 13];
        W1i_1p_b=[37 -13 13+w1len];
        W1i_2p_a=[37 13 -13];
        W1i_2p_b=[37 13 -13-w1len];

        W1b_1p_a=W1i_1p_a*ROT;
        W1b_1p_b=W1i_1p_b*ROT;
        W1b_2p_a=W1i_2p_a*ROT;
        W1b_2p_b=W1i_2p_b*ROT;
        line([W1b_1p_a(1) W1b_1p_b(1)],[W1b_1p_a(2)
W1b_1p_b(2)],[W1b_1p_a(3) W1b_1p_b(3)],'Color','g','LineWidth',3)
        line([W1b_2p_a(1) W1b_2p_b(1)],[W1b_2p_a(2)
W1b_2p_b(2)],[W1b_2p_a(3) W1b_2p_b(3)],'Color','g','LineWidth',3)
    end

    if O_dot(nn,3)>0
        w1len=abs(O_dot(nn,1))*w_scl;

        W1i_1n_a=[37 13 13];
        W1i_1n_b=[37 13 13+w1len];
        W1i_2n_a=[37 -13 -13];
        W1i_2n_b=[37 -13 -13-w1len];

        W1b_1n_a=W1i_1n_a*ROT;
        W1b_1n_b=W1i_1n_b*ROT;
        W1b_2n_a=W1i_2n_a*ROT;
        W1b_2n_b=W1i_2n_b*ROT;
        line([W1b_1n_a(1) W1b_1n_b(1)],[W1b_1n_a(2)
W1b_1n_b(2)],[W1b_1n_a(3) W1b_1n_b(3)],'Color','g','LineWidth',3)
        line([W1b_2n_a(1) W1b_2n_b(1)],[W1b_2n_a(2)
W1b_2n_b(2)],[W1b_2n_a(3) W1b_2n_b(3)],'Color','g','LineWidth',3)
    end

    % Axis 2
    if O_dot(nn,2)<0
        w2len=abs(O_dot(nn,2))*w_scl;

        W2i_1p_a=[37 0 13];
        W2i_1p_b=[37 0 13+w2len];
        W2i_2p_a=[-37 0 -13];
        W2i_2p_b=[-37 0 -13-w2len];

        W2b_1p_a=W2i_1p_a*ROT;
        W2b_1p_b=W2i_1p_b*ROT;
        W2b_2p_a=W2i_2p_a*ROT;
        W2b_2p_b=W2i_2p_b*ROT;
        line([W2b_1p_a(1) W2b_1p_b(1)],[W2b_1p_a(2)
W2b_1p_b(2)],[W2b_1p_a(3) W2b_1p_b(3)],'Color','g','LineWidth',3)
```

```
        line([W2b_2p_a(1) W2b_2p_b(1)],[W2b_2p_a(2)
W2b_2p_b(2)],[W2b_2p_a(3) W2b_2p_b(3)],'Color','g','LineWidth',3)
    end

    if O_dot(nn,2)>0
        w2len=abs(O_dot(nn,2))*w_scl;

        W2i_1n_a=[-37 0 13];
        W2i_1n_b=[-37 0 13+w2len];
        W2i_2n_a=[37 0 -13];
        W2i_2n_b=[37 0 -13-w2len];

        W2b_1n_a=W2i_1n_a*ROT;
        W2b_1n_b=W2i_1n_b*ROT;
        W2b_2n_a=W2i_2n_a*ROT;
        W2b_2n_b=W2i_2n_b*ROT;
        line([W2b_1n_a(1) W2b_1n_b(1)],[W2b_1n_a(2)
W2b_1n_b(2)],[W2b_1n_a(3) W2b_1n_b(3)],'Color','g','LineWidth',3)
        line([W2b_2n_a(1) W2b_2n_b(1)],[W2b_2n_a(2)
W2b_2n_b(2)],[W2b_2n_a(3) W2b_2n_b(3)],'Color','g','LineWidth',3)
    end

    % Axis 3
    if O_dot(nn,1)<0
        w3len=abs(O_dot(nn,3))*w_scl;

        W3i_1p_a=[37 -13 0];
        W3i_1p_b=[37 -13-w3len 0];
        W3i_2p_a=[-37 13 0];
        W3i_2p_b=[-37 13+w3len 0];

        W3b_1p_a=W3i_1p_a*ROT;
        W3b_1p_b=W3i_1p_b*ROT;
        W3b_2p_a=W3i_2p_a*ROT;
        W3b_2p_b=W3i_2p_b*ROT;
        line([W3b_1p_a(1) W3b_1p_b(1)],[W3b_1p_a(2)
W3b_1p_b(2)],[W3b_1p_a(3) W3b_1p_b(3)],'Color','g','LineWidth',3)
        line([W3b_2p_a(1) W3b_2p_b(1)],[W3b_2p_a(2)
W3b_2p_b(2)],[W3b_2p_a(3) W3b_2p_b(3)],'Color','g','LineWidth',3)
    end

    if O_dot(nn,1)>0
        w3len=abs(O_dot(nn,3))*w_scl;

        W3i_1n_a=[37 13 0];
        W3i_1n_b=[37 13+w3len 0];
        W3i_2n_a=[-37 -13 0];
        W3i_2n_b=[-37 -13-w3len 0];

        W3b_1n_a=W3i_1n_a*ROT;
        W3b_1n_b=W3i_1n_b*ROT;
        W3b_2n_a=W3i_2n_a*ROT;
        W3b_2n_b=W3i_2n_b*ROT;
        line([W3b_1n_a(1) W3b_1n_b(1)],[W3b_1n_a(2)
W3b_1n_b(2)],[W3b_1n_a(3) W3b_1n_b(3)],'Color','g','LineWidth',3)
        line([W3b_2n_a(1) W3b_2n_b(1)],[W3b_2n_a(2)
W3b_2n_b(2)],[W3b_2n_a(3) W3b_2n_b(3)],'Color','g','LineWidth',3)
    end

    % Text Placement
    % 'Target Attitude' and 'Time'
```

```
    text(50,-50,-92,['Target Attitude: \theta_1 = ' num2str(targ(nn,1))
', \theta_2 = ' num2str(targ(nn,2)) ', \theta_3 = '
num2str(targ(nn,3))],'FontWeight','bold','FontSize',11)
    text(8,-8,-75, 'Time:')
    text(-3,3,-75, num2str(dt*(nn-1)))

    tx_off=47;      % overall text offset - horizontal
    tx_off_z=90;    % overall text offset - vertical

    %Euler Deltas
    e_off_num=9;    % Euler Delta number offset - horizontal
    e_off_unit=21;  % Euler Delta unit offset - horizontal

    text(tx_off-e_off_num,-tx_off+e_off_num,tx_off_z,
num2str(round(10*(targ(nn,1)-th1*180/pi))/10),'FontSize',8)
    text(tx_off,-tx_off,tx_off_z, '\Delta\theta_1=','FontSize',8)
    text(tx_off-e_off_unit,-tx_off+e_off_unit,tx_off_z,
'deg','FontSize',8)

    text(tx_off-e_off_num,-tx_off+e_off_num,tx_off_z-10,
num2str(round(10*(targ(nn,2)-th2*180/pi))/10),'FontSize',8)
    text(tx_off,-tx_off,tx_off_z-10, '\Delta\theta_2=','FontSize',8)
    text(tx_off-e_off_unit,-tx_off+e_off_unit,tx_off_z-10,
'deg','FontSize',8)

    text(tx_off-e_off_num,-tx_off+e_off_num,tx_off_z-20,
num2str(round(10*(targ(nn,3)-th3*180/pi))/10),'FontSize',8)
    text(tx_off,-tx_off,tx_off_z-20, '\Delta\theta_3=','FontSize',8)
    text(tx_off-e_off_unit,-tx_off+e_off_unit,tx_off_z-20,
'deg','FontSize',8)

    %body rates
    b_off=-34;      % body rate overall offset - horizontal
    b_off_num=7;    % body rate number offset - horizontal
    b_off_unit=19;  % body rate unit offset - horizontal

    text(tx_off-b_off_num+b_off,-tx_off+b_off_num-b_off,tx_off_z,
num2str(round(100*(w1*60/(2*pi)))/100),'FontSize',8)
    text(tx_off+b_off,-tx_off-b_off,tx_off_z, '\omega_1=','FontSize',8)
    text(tx_off-b_off_unit+b_off,-tx_off+b_off_unit-b_off,tx_off_z,
'RPM','FontSize',8)

    text(tx_off-b_off_num+b_off,-tx_off+b_off_num-b_off,tx_off_z-10,
num2str(round(100*(w2*60/(2*pi)))/100),'FontSize',8)
    text(tx_off+b_off,-tx_off-b_off,tx_off_z-10,
'\omega_2=','FontSize',8)
    text(tx_off-b_off_unit+b_off,-tx_off+b_off_unit-b_off,tx_off_z-10,
'RPM','FontSize',8)

    text(tx_off-b_off_num+b_off,-tx_off+b_off_num-b_off,tx_off_z-20,
num2str(round(100*(w3*60/(2*pi)))/100),'FontSize',8)
    text(tx_off+b_off,-tx_off-b_off,tx_off_z-20,
'\omega_3=','FontSize',8)
    text(tx_off-b_off_unit+b_off,-tx_off+b_off_unit-b_off,tx_off_z-20,
'RPM','FontSize',8)

    %Euler Angles
    a_off=-68;      % Euler Angle overall offset - horizontal
    a_off_num=6;    % Euler Angle number offset - horizontal
    a_off_unit=19;  % Euler Angle unit offset - horizontal
```

```
    text(tx_off-a_off_num+a_off,-tx_off+a_off_num-a_off,tx_off_z,
num2str(round(10*(th1*180/pi))/10),'FontSize',8)
    text(tx_off+a_off,-tx_off-a_off,tx_off_z, '\theta_1=','FontSize',8)
    text(tx_off-a_off_unit+a_off,-tx_off+a_off_unit-a_off,tx_off_z,
'deg','FontSize',8)

    text(tx_off-a_off_num+a_off,-tx_off+a_off_num-a_off,tx_off_z-10,
num2str(round(10*(th2*180/pi))/10),'FontSize',8)
    text(tx_off+a_off,-tx_off-a_off,tx_off_z-10,
'\theta_2=','FontSize',8)
    text(tx_off-a_off_unit+a_off,-tx_off+a_off_unit-a_off,tx_off_z-10,
'deg','FontSize',8)

    text(tx_off-a_off_num+a_off,-tx_off+a_off_num-a_off,tx_off_z-20,
num2str(round(10*(th3*180/pi))/10),'FontSize',8)
    text(tx_off+a_off,-tx_off-a_off,tx_off_z-20,
'\theta_3=','FontSize',8)
    text(tx_off-a_off_unit+a_off,-tx_off+a_off_unit-a_off,tx_off_z-20,
'deg','FontSize',8)

    % figure properties
    axis equal
    axis([-50 50 -50 50 -50 50])
    view([10 10 10])

    %capture movie frame
    M(f) = getframe;

end
% mmm = input('Create movie file?','s');
% if mmm=='y'
%     movie2avi(M,'c:\temp\sim_mov','compression','Indeo5')
% end
```

```
% data_get

clc
close all

load C:\~users~\French\SIMSAT\mm_80_10.mat
data=mm_80_10

if th_target>0
    leg=4
else
    leg=1
end

tx_scale=th_target/45

ph_dif=1.5;

y1_data=data.Y(5).Data; %position
y2_data=data.Y(6).Data; %wheel
y3_data=data.Y(1).Data; %energy
y4_data=data.Y(4).Data; %thrust
t_data=data.X(1).Data;

% Settling Time Subroutine-----------------------
th_set_count_act=0;
th_set_count_mod=0;
count_stop_act=0;
count_stop_mod=0;
th_set_targ=th_target;
th_set_delta=abs(.05*th_target);

%actual
for a=length(t_data):-1:1;
    if count_stop_act==0
        if y1_data(a)*180/pi < th_set_targ+th_set_delta
            if y1_data(a)*180/pi > th_set_targ-th_set_delta
                th_set_count_act=th_set_count_act+1;
            else count_stop_act=1;
            end
        else count_stop_act=1;
        end
    end
end
T_set_act=(length(t_data)-1-th_set_count_act)*dt+ph_dif;

%model
for a=length(t_data):-1:1;
    if count_stop_mod==0
        if satang(a)*180/pi < th_set_targ+th_set_delta
            if satang(a)*180/pi > th_set_targ-th_set_delta
                th_set_count_mod=th_set_count_mod+1;
            else count_stop_mod=1;
            end
        else count_stop_mod=1;
        end
    end
end
T_set_mod=(length(t_data)-1-th_set_count_mod)*dt;

% ----------------------------------------------END
```

```
samp_1=1/dt;
samp_2=115/dt;

ph_dt_samp=ph_dif/dt;
%
% figure
% hold on
%
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),y3_data(samp_1:samp_2))
% plot(t_data,sim_energy(:,3),'r--')
% title('Energy')
% grid
% xlabel('sec')
% ylabel('E')
% legend('actual','model')

th_set_1=(th_set_targ-th_set_delta)*ones(tend/dt+1,1);
th_set_2=(th_set_targ+th_set_delta)*ones(tend/dt+1,1);

figure
subplot(3,1,1)
hold on
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),180/pi*y1_data(samp_1:s
amp_2))
plot(t_data,180/pi*satang(:,1),'r--')
plot(t_data,th_set_1,'g:')
plot(t_data,th_set_2,'g:')
title('Yaw Axis Position')
grid
% xlabel('sec')
ylabel('deg')
legend('actual','model','settling boundaries',leg)
text(60,32*tx_scale,['t_s_e_t_,_m_o_d =' num2str(T_set_mod) ' sec'])
text(60,(32-12*abs(tx_scale))*tx_scale,['t_s_e_t_,_a_c_t ='
num2str(T_set_act) ' sec'])

subplot(3,1,2)
hold on

plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),3*T*y4_data(samp_1:samp
_2))
plot(t_data(samp_1:samp_2),thrust(samp_1:samp_2,1),'r--')

title('Yaw Axis Thrust')
grid
% xlabel('sec')
ylabel('lbf')
legend('actual','model')
text(60,-.05,['T(+) =' num2str(3*T) ' lb'])
text(60,-.15,['T(-) =' num2str(3*T*T_bias) ' lb'])
T_f_act=sum(abs(y4_data))*dt;
T_f_mod=sum(sum(T_f_6))*dt;
text(90,-.05,['t_f_,_m_o_d =' num2str(T_f_mod) ' sec'])
text(90,-.15,['t_f_,_a_c_t =' num2str(T_f_act) ' sec'])

subplot(3,1,3)
hold on
O_scale=-32.2;
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),y2_data(samp_1:samp_2))
```

```
plot(t_data(samp_1:samp_2),O_scale*O_dot(samp_1:samp_2,1),'r--')

title('Yaw Axis Wheel Rate Change')
grid
xlabel('sec')
ylabel('deg/sec^2')
legend('actual','model',leg)

set(1,'Position',[600 10 500 750])

%
% figure
% subplot(2,1,1)
% hold on
%
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),180/pi*y1_data(samp_1:s
amp_2))
% plot(t_data,180/pi*satang(:,1),'r--')
% plot(t_data,th_set_1,'g')
% plot(t_data,th_set_2,'g')
% title('Yaw Axis Position')
% grid
% xlabel('sec')
% ylabel('deg')
% legend('actual','model',2)
% text(70,7,['T_s_e_t_,_m_o_d =' num2str(T_set_mod) ' sec'])
% text(70,-8,['T_s_e_t_,_a_c_t =' num2str(T_set_act) ' sec'])
%
%
% subplot (2,1,2)
% hold on
%
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),3*T*y4_data(samp_1:samp
_2))
% plot(t_data,thrust(:,1),'r--')
%
% title('Yaw Axis Thrust')
% grid
% xlabel('sec')
% ylabel('lbf')
% legend('actual','model')
%
% T_f_act=sum(abs(y4_data))*dt;
% T_f_mod=sum(sum(T_f_6))*dt;
% text(70,-.05,['T_f_,_m_o_d =' num2str(T_f_mod) ' sec'])
% text(70,-.15,['T_f_,_a_c_t =' num2str(T_f_act) ' sec'])
%
% figure
% subplot(2,1,1)
% hold on
%
%
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),3*T*y4_data(samp_1:samp
_2))
% plot(t_data(samp_1:samp_2),thrust(samp_1:samp_2,1),'r--')
%
% title('Yaw Axis Thrust')
% grid
% xlabel('sec')
% ylabel('lbf')
% legend('actual','model')
```

```
%
% subplot(2,1,2)
% hold on
% O_scale=-40;
%
plot(t_data(samp_1+ph_dt_samp:samp_2+ph_dt_samp),y2_data(samp_1:samp_2))
% plot(t_data(samp_1:samp_2),O_scale*O_dot(samp_1:samp_2,1),'r--')
%
% title('Yaw Axis Wheel Rate Change')
% grid
% xlabel('sec')
% ylabel('deg/sec^2')
% legend('actual','model')
```

Appendix C: Data Tables

th_target	Target Orientation (deg)	20	-20	40	-40	60	-60	80
Control Variables		**Values**						
ep_K_T	thruster dead band	0.05	0.05	0.05	0.05	0.05	0.05	0.05
K_r_T	thruster rate gain	10	10	10	10	10	10	10
K_d_T	thruster delta gain	1	1	1	1	1	1	1
K_o_T	thruster overall gain	1	1	1	1	1	1	1
K_r_W	wheel rate gain	10000	10000	10000	10000	10000	10000	10000
K_d_W	wheel delta gain	1000	1000	1000	1000	1000	1000	1000
K_o_W	wheel overall gain	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1
eswitch	control switch energy	0.015	0.015	0.015	0.015	0.015	0.015	0.015
System Match Settings								
K_motor	motor out gain	0.1	0.1	0.1	0.1	0.1	0.1	0.1
K_motor_in	motor in gain	1	1	1	1	1	1	1
T_pos	thurst in positive direction per sole	0.045	0.045	0.035	0.045	0.042	0.045	0.07
T_bias	thrust bias (negative stronger)	1.25	1.25	1.25	1.25	1.25	1.25	1.25
Performance Measures								
Model								
T_s (sec)	Settling Time [5% criteria] (sec)	43.85	42.1	50.2	47.45	53.5	50.6	53.05
T_f (sec)	Thrust Time (sec)	10.35	8.5	20.05	17.4	24.6	24.7	21.65
Actual								
T_s (sec)	Settling Time [5% criteria] (sec)	43.1	41.6	50.85	47.95	53.45	50.7	53.55
T_f (sec)	Thrust Time (sec)	8.6	8.3	16.9	16.35	21.2	21.85	15.15

Table 4. Constant gain – varied angle data

th_target	Target Orientation (deg)	45	45	45	45	45	45	45	45	45	45	45	45	45	45
Control Variables															
ep_K_T	thruster dead band	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05	0.05
K_r_T	thruster rate gain	5	6	7	8	9	10	11	12	13	14	15	20	25	30
K_d_T	thruster delta gain	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K_o_T	thruster overall gain	1	1	1	1	1	1	1	1	1	1	1	1	1	1
K_r_W	wheel rate gain	5000	6000	7000	8000	9000	10000	11000	12000	13000	14000	15000	20000	25000	30000
K_d_W	wheel delta gain	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000
K_o_W	wheel overall gain	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1	-1.1
eswitch	control switch energy	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015	0.015
System Match Settings															
K_motor	motor out gain	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
K_motor_in	motor in gain	1	1	1	1	1	1	1	1	1	1	1	1	1	1
T_bias	thrust bias (negative stronger)	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25	1.25
Performance Measures															
T_av=.035															
T_s (sec)	Settling Time [5% criteria] (sec)	63.6	58.45	43.6	48.15	53.55	56.6	59.2	61.7	64.25	66.85	69.45	83	97.1	111.45
T_f (sec)	Thrust Time (sec)	42.65	36	26.1	26.65	25.6	24.5	23.5	22.6	21.75	20.9	20.05	16.75	14.25	12.35
T_av=.045															
T_s (sec)	Settling Time [5% criteria] (sec)	55.6	47	44.25	49.8	52.75	55.3	57.85	60.45	63.05	65.7	68.4	82.25	96.65	111.3
T_f (sec)	Thrust Time (sec)	33.7	27.25	23.65	22.55	21.5	20.45	19.55	18.7	17.85	17.1	16.45	13.45	11.35	9.8
T_av=.055															
T_s (sec)	Settling Time (5%)	49.15	39.85	45.95	49.27	51.85	54.4	56.95	59.6	62.3	65	67.75	81.9	96.45	111.35
T_f (sec)	Thrust Time	27.4	21.15	20.65	19.55	18.55	17.7	16.75	16	15.25	14.6	13.95	11.35	9.55	8.2

Table 5. Constant angle – varied gain data excerpt

.

Bibliography

Beda, Peter B. "Anticipatory Computing in the Attitude Control of Satellites." Computing Anticipatory Systems: CASSYS 2000 – Fourth International Conference. (PLACE OF PUBLICATION): American Institute of Physics. 2001.

"Boeing Satellite Systems." Online Catalogue. http://www.boeing.com/defense-space/space/bss/flash.html. 14 January 2003.

Colebank, James E. and others. SIMSAT: *A Satellite System Simulator and Experimental Test Bed for Air Force Research*. MS Thesis, AFIT/GSE/GSO/ENY/99M-1. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, March 1999.

Chobotov, Vladimir A. *Spacecraft Attitude Dynamics and Control.* Malabar, Florida: Krieger Publishing Company, 1991.

Dabrowski, Vincent J. *Experimental Demonstration of an Algorithm to Detect the Presence of a Parasitic Satellite.* MS Thesis, AFIT/GAE/ENY/03M-2. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, March 2003.

Dempster, M. A. H. and G. M. Coupé. "Investigation of the Stability of Satellite Large Angle Attitude Manoeuvers Using Nonlinear Optimization Methods." *Automatic Control in Space 1982*. Oxford: Pergamon Press. July 1982.

"Driving Hubble Through Space". Online Article. http://hubble.nasa.gov/hubble-operations/driving-hubble.html. 10 January 2003.

Fischer, Steven B. and others. *Satellite Integrated Power and Attitude Control System Design Study*. MS Thesis, AFIT/GSE/GSO/ENY/97D-1. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, December 1997.

Frazzoli, Emilio. *Robust Hybrid Control for Autonomous Vehicle Motion Planning.* Doctoral Dissertation. Massachusetts Institute of Technology, Boston, Massachusetts, June 2001.

Fulton, Joseph M. *Attitude Control and Multimedia Representation of Air Force Institute of Technology's (AFIT's) Simulation Satellite (SIMSAT).* MS Thesis, AFIT/GA/ENY/00M-01. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, March 2000.

Hewston, Alan W., Kent A. Mitchell and Jerzy T. Sewicki. "Attitude Control Subsystem for the Advanced Communications Technology Satellite." NASA Technical Memorandum 107352. November 1996.

"Intelsat 7, 7A." Article in Jet Propulsion Laboratory Online Library. http://samadhi.jpl.nasa.gov/msl/QuickLooks/intelsat77aQL.html. 14 January 2003.

Irvin, David J. *A Study of Linear vs. Nonlinear Control Techniques for the Reconfiguration of Satellite Formations.* MS Thesis, AFIT/GA/ENY/01M-02. School of Engineering and Management, Air Force Institute of Technology (AU), Wright-Patterson AFB, Ohio, March 2001.

Jane's Space Directory. Alexandria, VA: Jane's Information Group, Ltd. 2002.

Kooy, J. M. J. "On Automatic Internal and External Control of Long Range Rockets," *Proceedings of the 5th International Congress on Astronautics*. Innsbruck, Austria: International Astronautical Congress. 1955.

Levinthal, J. and others. "Space Platform Attitude Control System." *Automatic Control in Space 1982*. Oxford: Pergamon Press. July 1982.

McDuffie, J. H. and Yu B. Shtessel. "A Decoupled Sliding Mode Controller and Observer for Satellite Attitude Control," *Proceedings of the 1997 American Control Conference*. 564-564. Evanston, IL: American Automatic Control Council, 1997.

Ogata, Katsuhiko. *Modern Control Engineering* (4th Edition). Upper Saddle River, New Jersey: Prentice Hall, 2002.

Parmalee, Patricia. "Cancel the Hubble Valentine," *Aviation Week and Space Technology*, 155: 25 (December 24-31, 2001).

Parrish, David K. *Applications of Nonlinear Control Using the State-Dependent Riccati Equation*. MS thesis, AFIT/GA/ENY/95D-02. School of Engineering and Management, Air Force Institute of Technology (AETC), Wright-Patterson AFB OH, September 1994.

"Quick Facts." Online Fact Sheet. http://hubblesite.org/sci.d.tech/facts_.and._figures/quick_facts/quick_facts_3.shtml. 15 Jan 02.

Raushenbakh, B. V. "The First Control System for Space Vehicles," IAF PAPER 89-728. *Proceedings of the 40th International Astronautical Congress*. Malaga, Spain. 1989.

Sedov, L. I. "Ten Years of Space Exploration by the Soviet Union". *Proceedings of the 18th International Astronautical Conference*. Belgrade: Pergamon Press. 1967.

Skaar, Steven Baard. *Arbitrary Large-Angle Satellite Attitude Maneuvers Using an Optimal Reaction Wheel Power Criterion*. PhD dissertation. Virginia Polytechnic Institute and State University, Blacksburg, VA, August 1982.

"Soviet Satellites." *The McGraw-Hill Encyclopedia of Space*. Paris: The Hamlyn Publishing Group Ltd. 1968.

St. Pierre, Daniel C. *Satellite Attitude Dynamics Simulation and Control (SADSaC).* MS Thesis. University of Colorado, Colorado Springs, CO, 4 May 1998.

Thomas, Michael G. *Photovoltaic Systems for Government Agencies.* Technical Report for Sandia National Laboratories (US Dept of Energy).

TRW Space Log 1970-71. Redondo Beach, California: TRW Systems Group. 1971.

de Weck, Oliver L. "Attitude Determination and Control (ADCS)." Online presentation. http://web.mit.edu/16.83/www/lectures/ACS.pdf. 17 January 03.

Wertz, James R. *Mission Geometry; Orbit and Constellation Design and Management*. El Segundo, California: Microcosm Press and Kluwer Academic Publishers. 2001.

-----. *Spacecraft Attitude Determination and Control.* London: D. Reidel Publishing Company.

-----. *Space Mission Analysis and Design* (2nd Edition). Torrance, California: Microcosm, Inc., 1992.

Wie, Bong. *Space Vehicle Dynamics and Control.* Reston, Virginia: American Institute for Aeronautics and Astronautics (AIAA), 1998.

Wiesel, William E. *Spaceflight Dynamics* (2nd Edition). Boston: Irwin McGraw-Hill. 1997.

Vita

Capt David B. French graduated from the University of Tennessee in Knoxville in June of 1993 and was commissioned via Officer Training School at Maxwell Air Force Base, Alabama, on 5 May 1995.

Capt French was assigned to Hurlburt Field, Florida as a maintenance officer in the 8th Special Operations Squadron, 16th Special Operations Wing, where he supervised the maintenance of six MC-130E Combat Talon I aircraft. After 6 months, Captain French was transferred to the 16th Aircraft Generation Squadron, where he served as the MC-130E Aircraft Maintenance Unit Officer-In-Charge. In October of 1997, he was transferred to the 16th Equipment Maintenance Squadron, assuming command of the Fabrication Flight. While at Hurlburt Field, Captain French was deployed to Exercise BRIGHT STAR, Cairo, Egypt, in 1995, NATO Exercise TRIAL BENDBOW, Cold Lake, Canada, in 1996, Operation SOUTHERN WATCH, Prince Sultan Air Base, Saudi Arabia, in 1997, as well as numerous CONUS deployments.

In 1998, Captain French was reassigned to Maxwell AFB, Alabama, as a Flight Training Officer for the 24th Training Squadron, Officer Training School. In July of 1999, he was promoted to Deputy Flight Commander, D-Flight. In March of 2000, he was tasked with "standing up" a new flight for the expanding school, E-Flight. He then served as E-Flight commander until his transfer to AFIT in August of 2001.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 074-0188

The public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of the collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to an penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number.
PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.

1. REPORT DATE *(DD-MM-YYYY)*	2. REPORT TYPE	3. DATES COVERED *(From – To)*
25-03-2003	**Master's Thesis**	Jun 2002-March 2003

4. TITLE AND SUBTITLE

HYBRID CONTROL STRATEGIES FOR RAPID, LARGE ANGLE SATELLITE SLEW MANEUVERS

5a. CONTRACT NUMBER

5b. GRANT NUMBER

5c. PROGRAM ELEMENT NUMBER

6. AUTHOR(S)

French, David B., Captain, USAF

5d. PROJECT NUMBER

5e. TASK NUMBER

5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAMES(S) AND ADDRESS(S)

Air Force Institute of Technology
Graduate School of Engineering and Management (AFIT/EN)
2950 Hobson Way, Building 642
WPAFB OH 45433-7765

8. PERFORMING ORGANIZATION REPORT NUMBER

AFIT/GA/ENY/03-2

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSOR/MONITOR'S ACRONYM(S)

11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED.

13. SUPPLEMENTARY NOTES

14. ABSTRACT

This research investigated hybrid control strategies for rapid satellite pointing. First, a detailed computer simulation model of AFIT's SIMSAT satellite simulator was constructed. Control strategies were developed to enable the system to perform wide-angle, 3-axis slewing maneuvers using a combination of both thrusters and reaction wheels. To handle the non-linear model, a state-dependent Riccati equation (SDRE) controller was programmed and successfully controlled the computer-modeled satellite for any given slewing maneuver. A simpler PD controller was then programmed and demonstrated on the computer simulation of the SIMSAT, using a combination of thruster and reaction wheel control inputs for wide-angle single axis maneuvers and for small angles using three-axis control. There was good agreement between the experimentally obtained maneuver results and those produced with the computer simulation model for the single-axis case. Lastly, the trade-off between settling time and thruster fuel is discussed, as well as the variation of gains required to achieve maximum performance given an angular goal.

15. SUBJECT TERMS

SIMSAT, satellite slew, large angle maneuver, State Dependent Riccati Equation (SDRE), Linear Quadratic Regulator (LQR), cold gas thrusters, reaction wheels, satellite test bed, hybrid control, control switching

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT	b. ABSTRACT	c. THIS PAGE			Richard G. Cobb, Maj, USAF (ENY)
U	U	U	UU	135	19b. TELEPHONE NUMBER *(Include area code)* (937) 255-6565, ext 4669; e-mail: Richard.Cobb@afit.edu

Standard Form 298 (Rev. 8-98)
Prescribed by ANSI Std. Z39-18